# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# DISSERTATION

**FEASIBILITY STUDY OF AN AERIAL MANIPULATOR INTERACTING WITH A VERTICAL WALL**

by

Bruno G. F. Tavora

June 2017

| | |
|---|---|
| Dissertation Supervisor: | Xiaoping Yun |

**Approved for public release. Distribution is unlimited.**

THIS PAGE INTENTIONALLY LEFT BLANK

| | |
|---|---|
| **REPORT DOCUMENTATION PAGE** | *Form Approved OMB No. 0704-0188* |

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

| | | |
|---|---|---|
| **1. AGENCY USE ONLY**<br>***(Leave blank)*** | **2. REPORT DATE**<br>June 2017 | **3. REPORT TYPE AND DATES COVERED**<br>Dissertation |

**4. TITLE AND SUBTITLE**
FEASIBILITY STUDY OF AN AERIAL MANIPULATOR INTERACTING WITH A VERTICAL WALL

**5. FUNDING NUMBERS**

**6. AUTHOR(S)** Bruno G. F. Tavora

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Naval Postgraduate School
Monterey, CA 93943-5000

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
N/A

**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB number ____N/A____.

**12a. DISTRIBUTION / AVAILABILITY STATEMENT**
Approved for public release. Distribution is unlimited.

**12b. DISTRIBUTION CODE**

**13. ABSTRACT (maximum 200 words)**

Modeling and control of aerial manipulators are presented in this study. An aerial manipulator is an unmanned aerial vehicle (UAV) equipped with a robotic arm. Aerial manipulators have potential for many applications, such as rescue operations along tall buildings and high cliffs, maintenance and repairing of power line equipment, and retrieval of items from difficult-to-access locations. Control of aerial manipulators is a challenging problem due to dynamic interactions between aerial vehicles and robotic arms. Additionally, when an aerial manipulator operates near or on a vertical surface such as a wall, there is the near-wall effect, which is the aerodynamic disturbance caused by the proximity of the flying vehicle to a wall. One of the objectives of this dissertation is to study the near-wall effect and develop controllers that are able to mitigate this disturbance when they are flying close to a wall. Another objective is to develop a controller to allow an aerial manipulator to fly in close proximity to a vertical surface and perform manipulation tasks by interacting with the surface to achieve the desired interaction forces and torques. Nonlinear models of generic aerial manipulators, as well as of a prototype aerial manipulator composed of a hexacopter with a three-joint robotic arm, are established. The near-wall effect is characterized by laboratory flight experiments. A free-flight controller incorporating the near-wall effect is proposed to compensate the near-wall disturbance. An equilibrium-based force/torque controller is developed for executing tasks that require the aerial manipulator to exert forces and torques on a wall. The simulation and experimental results validated the performance of the controllers with successful near-wall flying while applying a specified amount of forces and torques on an object fixed on a wall.

**14. SUBJECT TERMS**
unmanned aerial vehicle, multicopter, hexacopter, aerial manipulator, robotic arm, robotic manipulator, near-wall effect, kinematics, dynamics, nonlinear control.

**15. NUMBER OF PAGES**
135

**16. PRICE CODE**

| | | | |
|---|---|---|---|
| **17. SECURITY CLASSIFICATION OF REPORT**<br>Unclassified | **18. SECURITY CLASSIFICATION OF THIS PAGE**<br>Unclassified | **19. SECURITY CLASSIFICATION OF ABSTRACT**<br>Unclassified | **20. LIMITATION OF ABSTRACT**<br>UU |

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

# FEASIBILITY STUDY OF AN AERIAL MANIPULATOR INTERACTING WITH A VERTICAL WALL

Bruno G. F. Tavora
Major, Brazilian Air Force
B.S., Technological Institute of Aeronautics, Brazil, 2003

Submitted in partial fulfillment of the
requirements for the degree of

**DOCTOR OF PHILOSOPHY IN ELECTRICAL ENGINEERING**

from the

**NAVAL POSTGRADUATE SCHOOL**
**June 2017**

Approved by: Xiaoping Yun
Distinguished Professor of Electrical and Computer Engineering
Dissertation Supervisor and Dissertation Committee Chair

Marcello Romano
Professor of Mechanical
and Aerospace Engineering
Dissertation Co-Advisor

Herschel H. Loomis
Distinguished Professor of Electrical
and Computer Engineering

Roberto Cristi
Professor of Electrical
and Computer Engineering

Robert G. Hutchins
Associate Professor of Electrical
and Computer Engineering

Approved by: R. Clark Robertson
Chair, Department of Electrical and Computer Engineering

Approved by: Douglas Moses
Vice Provost of Academic Affairs

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

Modeling and control of aerial manipulators are presented in this study. An aerial manipulator is an unmanned aerial vehicle (UAV) equipped with a robotic arm. Aerial manipulators have potential for many applications, such as rescue operations along tall buildings and high cliffs, maintenance and repairing of power line equipment, and retrieval of items from difficult-to-access locations. Control of aerial manipulators is a challenging problem due to dynamic interactions between aerial vehicles and robotic arms. Additionally, when an aerial manipulator operates near or on a vertical surface such as a wall, there is the near-wall effect, which is the aerodynamic disturbance caused by the proximity of the flying vehicle to a wall. One of the objectives of this dissertation is to study the near-wall effect and develop controllers that are able to mitigate this disturbance when they are flying close to a wall. Another objective is to develop a controller to allow an aerial manipulator to fly in close proximity to a vertical surface and perform manipulation tasks by interacting with the surface to achieve the desired interaction forces and torques. Nonlinear models of generic aerial manipulators, as well as of a prototype aerial manipulator composed of a hexacopter with a three-joint robotic arm, are established. The near-wall effect is characterized by laboratory flight experiments. A free-flight controller incorporating the near-wall effect is proposed to compensate the near-wall disturbance. An equilibrium-based force/torque controller is developed for executing tasks that require the aerial manipulator to exert forces and torques on a wall. The simulation and experimental results validated the performance of the controllers with successful near-wall flying while applying a specified amount of forces and torques on an object fixed on a wall.

THIS PAGE INTENTIONALLY LEFT BLANK

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

# LIST OF ACRONYMS AND ABBREVIATIONS

| | |
|---|---|
| CAN | controller area network |
| DAQ | data acquisition |
| DoF | degrees of freedom |
| GPS | global positioning system |
| I2C | inter-integrated circuit |
| IBVS | image-based visual servoing |
| IDA-PBC | damping assignment passivity-based control |
| LiPo | lithium polymer |
| MPC | model predictive control |
| MRPC | model reference predictive control |
| PD | proportional derivative |
| PID | proportional integral derivative |
| PWM | pulse width modulation |
| UART | universal synchronous receiver/transmitter |
| UAV | unmanned aerial vehicle |
| UDP | user datagram protocol |
| USB | universal serial bus |
| VPIB | variable parameter integral backstepping |

THIS PAGE INTENTIONALLY LEFT BLANK

# ACKNOWLEDGMENTS

I would like to thank my wife, Tatiana, who interrupted her professional career to support mine, for taking care of our family while I was studying. I would like to thank my advisor, Professor Xiaoping Yun, for his commitment, patience, and guidance. I would like to thank my co-advisor, Professor Marcello Romano, for the opportunity to work at the Spacecraft Robotics Laboratory. I would like to thank Professors Herschel Loomis, Robert Hutchins, and Roberto Cristi for their work as members of the dissertation committee. I would like to thank Dr. Hyeongjun Park and all the folks at the Spacecraft Robotics Laboratory for their help while we were preparing the experiments. I would like to thank Dr. Kevin Jones, who built the hexacopter used in this research. He was patient and helpful while Dr. Hyeongjun Park and I were learning how to operate and fix the hexacopter. I would like to thank Professor James Calusdian for his voluntary help in several moments during this research.

THIS PAGE INTENTIONALLY LEFT BLANK

# I. INTRODUCTION

Unmanned aerial vehicle (UAV) research has been increasing progressively in the past decade for military and civilian applications. For some applications, the use of manned aircraft is unacceptable due to the high risks for the pilot. For instance, UAVs have been deployed in high-risk operations to perform aggressive maneuvers for obstacle avoidance or to collect research data in dangerous environments like hurricanes. UAVs can also be smaller, since they do not have to carry the weight of a human body. They are potentially less expensive and suitable to operate in smaller spaces, which makes them appropriate also for indoor applications; therefore, UAVs have become a versatile class of aerial vehicles for a number of applications, such as aerial reconnaissance, load transportation, inspection, and law enforcement.

Multicopters, an important class of UAVs, have emerged during the last several years. A multicopter is a rotorcraft with more than two rotors. With the advancement of computer processing technology, better inertial sensors, and lithium-ion polymer (LiPo) batteries, multicopters can be effectively built. Since these vehicles have a very simple mechanical structure with no variable pitch propellers and can be designed with the appropriate size and number of rotors, they can be utilized as a simple and versatile solution for many applications. Recreational use of radio-controlled multicopters has become popular. It is now easier and cheaper to buy small multicopters and accessories, including sophisticated microcontrollers with large support through tutorials and internet-based forums for users and developers. These are some reasons why the interest in multicopter research for civilian and military applications has been growing rapidly. In the academic field, now that the problem of achieving basic flying is solved, researchers have turned their attention on more complex tasks, like aggressive maneuvering, trajectory tracking, obstacle avoidance, vision-based navigation, cooperative tasks, and aerial manipulation.

The applications of UAVs are usually limited to the exchange of information for tasks without physical interaction such as tracking, surveillance, mapping, and visual inspection. The use of UAVs equipped with robotic arms, capable of interacting with the environment, can significantly increase their capability. In recent years, a new area of research has been growing on the use of air manipulators for a larger variety of applications such as physical inspection, maintenance, cleaning walls, and collecting objects in areas of difficult access.

An important problem that has not been addressed by the academic community is the interaction of a multicopter equipped with a robotic arm and a vertical wall. There are several applications that necessitate the study of this problem, like performing maintenance on a piece of a vertical equipment, cleaning a wall, opening a door knob, or rescuing from a tall building. There were some research studies involving interaction with objects on the ground or multicopters with simple prismatic joints pushing a vertical wall; however, there is no record in the literature of an aerial manipulator with a more complex robotic arm, capable of producing forces or torques about any direction under interaction with a vertical obstacle.

Any task executed by an aerial manipulator on a vertical wall is accomplished by applying a combination of torques and forces. The objective of this dissertation is to investigate the feasibility of controlling a multicopter equipped with a multi-joint robotic arm to interact with a vertical wall while applying a specified force or torque. The strategies for controlling aerial manipulators, as well as multicopters under wall interaction, are investigated.

The aerial manipulation is divided in two levels. The path planning level generates the desired kinematic trajectory, and the execution level provides the inputs commands to make the vehicle to follow the reference trajectory while applying the desired interaction forces. The scope of this dissertation is restricted to the execution level.

In this dissertation, the research on the modeling and control of aerial manipulators, taking advantage of a modern laboratory facility, a multicopter equipped with a three-link robotic arm, and a force/torque sensor, are described. In Chapter II, a literature review is carried out to present the state of the art in aerial manipulation and other areas related to this research, like near-wall disturbances, multicopter control, and multicopters interacting with walls. Later in Chapter III, a mathematical model for the aerial manipulator used in this study is developed by considering the near-wall disturbance and the force interaction with an object fixed on the wall. In Chapter IV, the system identification methodology and experiments for a multicopter with six rotors (hexacopter) equipped with a robotic arm are detailed. In Chapter V, the proposed control solutions for the free-flight stage, when the vehicle approaches a wall, and for the wall interaction, are described. In Chapter VI, the simulator development is described and simulation results are presented for the aerial manipulator employed in the experiments, which are described in Chapter VII. The conclusions are presented in Chapter VIII.

THIS PAGE INTENTIONALLY LEFT BLANK

# II. BIBLIOGRAPHIC REVIEW

This literature review is divided in three parts. First, the state of the art of the aerial manipulation is detailed, followed by a description of the most recent advances in aerial interaction with a vertical wall, and finally several preliminary studies on near-wall aerodynamic disturbance are presented.

In [1], a dynamic model was derived from the Euler-Lagrangian formalism for a multicopter equipped with an *n*-DoF (degrees of freedom) manipulator arm. The generalized coordinates are the center of mass position of the multicopter, its rigid body orientation, and the robotic-arm-joints angles, which provides 6+n degrees of freedom to the system. A Cartesian impedance controller was proposed to control the end-effector position and orientation. The inertial, body-fixed, and end-effector-fixed frames for a typical aerial manipulator are shown in Figure 1. The authors presented a mathematical formulation that converts Cartesian forces and torques acting on the gripper to forces and torques acting on the generalized coordinate space, which enables an impedance control law to be designed in terms of the Lagrangian equation. The stability of the controlled system was proven by Lyapunov theory; however, the authors did not address the singularities that occur because multicopters are underactuated systems. Two simulation cases were studied to successfully validate the theoretical results. In the first case, an aerial manipulator was supposed to hover under simulated wind disturbance. In the second case, a constant force was applied to the end-effector to simulate physical interaction with a wall. An interesting characteristic of the Cartesian impedance control is the ability to increase the accuracy of the end-effector position even when the multicopter is moved by disturbance forces from its desired position.

An aerial manipulator with several links is a redundant system since there are several postures that result in the same end-effector position and orientation. In [2], a methodology to take advantage of aerial manipulator redundancy to perform secondary tasks was presented. The paper shows how to compute a matrix that projects the control law inputs for secondary tasks into the null space of the primary task control law.

Simulations show that secondary tasks, like making the vehicle hover in the same position, can be successfully controlled while the end-effector desired trajectory is achieved.

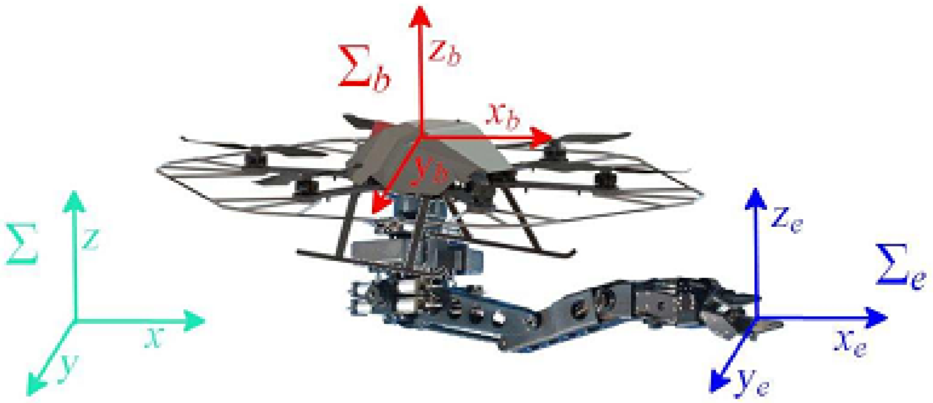


Figure 1. Multicopter and Robotic Arm. Source: [3].

A multicopter is known as an underactuated system since, from its six degrees of freedom, only four (vertical motion, roll, pitch, and yaw) are directly controlled, and the desired motion in the horizontal plane is achieved by controlling the attitude. In [4] and [5], a two-layer controller was presented, where the outer layer outputs the roll and pitch reference angles and the thrust force to control the vehicle position. The inner layer outputs the torques to control the attitude and robotic arm joints angles. Additionally, an inverse differential kinematics scheme was described to provide the generalized coordinates reference levels to result in the desired end-effector trajectory. In [5], an impedance filter is included in the trajectory control layer to make the vehicle compliant to external forces acting on the end-effector. Using a hierarchical approach for redundant systems, as in [2], primary and secondary tasks are described so as to make the vehicle compliant only under specific conditions. A simulation is presented in which the multicopter body is compliant with respect to the forces and torques on the vertical axis and the robotic arm is compliant in the horizontal plane. In [3], another extension of the controller presented in [4] considers model uncertainties by adding a disturbance term to the dynamic equation of the system and its estimation to the control law. Then, this adaptive controller is proven to improve the performance of a simulated system with model uncertainties.

In recent years, researchers from Drexel University have also started to research the use of robotic arms for aerial manipulation. In [6], a quadrotor with three manipulators of two degrees of freedom each is described. The vehicle and one of the arms are illustrated in Figure 2. To simplify the complex dynamics of the system, the flight is studied in two stages. In the first stage, the robotic arms are static while the multicopter is flying, while in the second stage, the multicopter is supposed to hover at a fixed position wile the manipulators move and perform manipulation on the environment. Simulated results show that it is possible to control an aerial manipulator accurately when the arms are moving and interacting with the ground.

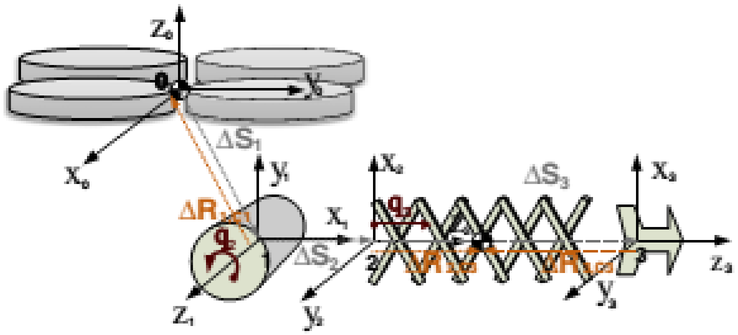


Figure 2. Quadcopter and One of the Manipulator Arms. Source: [6].

In [7], a gantry system is used to emulate the movements of a flying UAV and perform hardware-in-the-loop tests. As shown in Figure 3, the gantry system reproduces the movement that would be expected from the modeled UAV. The host platform is then equipped with a hyper-redundant robotic arm to perform manipulation tasks. Because the arm is hyper-redundant, secondary tasks can be performed and the end-effector can be controlled to have better accuracy than the host platform. Additionally, the controller is designed to make the robotic arm compliant with the manipulation interaction and increase the host platform stability. Simulations and experiments are described for the manipulator moving a perforated block from one post to another, lifting it from the first post, moving to a position aligned to the top of the second one, and finally sliding the block down. The same experimental setup is described in [8] to insert a hose into a pump. For this task, an impedance filter is implemented to provide compliance to the arm with respect to the manipulation forces. In [9], the hyper-redundant robotic arm is equipped with an *eye-in-hand* camera, and the end-effector is controlled to keep the same relative position with respect to a static object. The image-based visual servoing (IBVS) technique is implemented to translate the image feature position error to desired camera movements, and the host platform is controlled to minimize static torque from the weight of the robotic arm and to maximize the end-effector reachability.

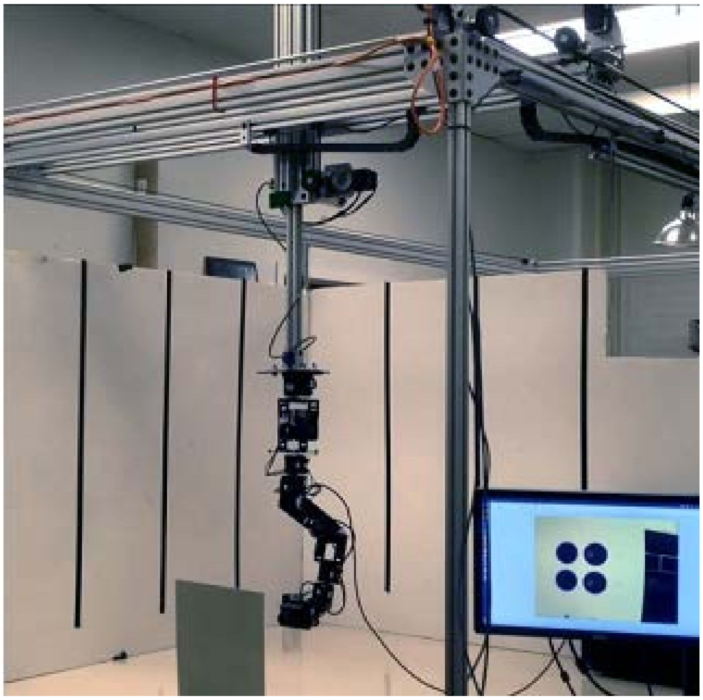


Figure 3. Gantry System to Emulate a Multicopter UAV. Source: [9].

Flight experiments are described in [10], where a quadcopter, equipped with two robotic arms, executes manipulation tasks. The paper improves a PID controller by means of a gain schedule technique, in which the gains are defined as a function of the robotic arms joints angles. Additionally, a Lyapunov-based model reference predictive control (MRPC) is combined with the gain schedule approach to overcome the inconsistency of some parameters that vary during flight as a result of changes in battery level, aerodynamic effect, arm pose, and the manipulated load. Finally, a Hybrid Automaton is proposed to switch states between arm deployment phase, manipulation phase, and adaptation phase. In [11], a similar aerial manipulator, with two robotic arms, is employed to turn a valve. The multicopter trajectory and robotic arms are commanded by means of a human-machine interface with support of a motion capture system. A PID controller is implemented for attitude stabilization. Different dynamic models are derived for free flight and for the valve turning task, where aerodynamic effects are substituted by valve turning friction terms. In Figure 4, the quadrotor rotates while the two robotic arms are holding the valve.

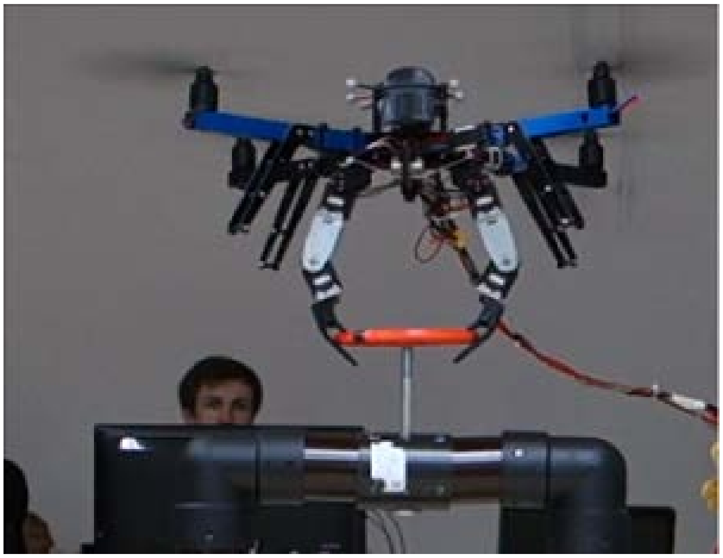


Figure 4. Valve Turning Experiment. Source: [11].

Simulations and experiments are performed in [12] to study the attitude control for a quadrotor with a three link robotic arm. A variable parameter integral backstepping (VPIB), with feedforward compensation for the robotic arm static torques is compared to a simple PID controller. In a VPIB controller, the system is modeled in terms of subsystems, which are connected in a multi-layered structure. The backstepping controller is designed from the inner layer to the outer layer to guarantee stability. In the test, the robotic arm is initially pointing down when it starts to move to a final position where it is pointing forward. During this movement, the VPIB controller is demonstrated to have smaller position errors than a PID controller. Rather than deriving the dynamic equations for the multicopter and robotic arm separately, in [13], the dynamic equations are derived for the full system. So, the same backstepping approach is implemented, but now static and dynamic torques produced by the robotic arm are considered for the computation of the attitude control law. The simulations and experiments confirm a better performance of the VPIB controller when compared to the simple PID implementation. Another solution for the aerial manipulation problem, an interconnection and damping assignment passivity-based control (IDA-PBC), is presented in [14]. In the IDA-PBC approach, the system is modeled from the Hamiltonian Mechanics in terms of energy-like functions. Then, a control law is designed to shape the energy terms of the closed-loop system so that, by means of Lyapunov theory, the convergence is guaranteed. The simulated case shows that the controlled system is robust to disturbances while converging to desired trajectories. In [15] and [16], a momentum-based external forces estimator is presented to estimate aerodynamic effects, like wind, and the forces and torques that are induced on the center of mass of the multicopter by  the robotic arm movement. The momentum-based estimator proposes a linear relationship between the real and estimated force in the Laplace domain, so that the estimated force converges to the real value. Applying this equation to the system's dynamic equation, the estimated force is expressed in terms of the generalized momentum vector. Experiments show that this approach successfully improves the performance of a multicopter flying in an environment with wind. As shown in Figure 5, the external forces estimator is also validated for an aerial manipulator with a moving the robotic arm.

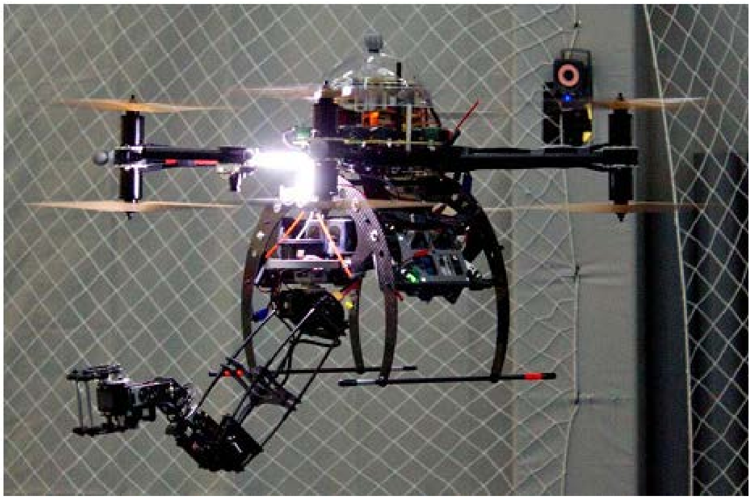


Figure 5. UAV Equipped with a Servo Robot Arm. Source: [16].

In [17] the authors prove that the dynamics of an aerial manipulator with a multi-joint robotic arm is decoupled into translational dynamics of the system's center of mass and internal rotational dynamics relative to the joints angles and the vehicle orientation Euler angles. Once the rotational and translational dynamics are decoupled, separate controllers are designed for each case, which is simpler. For validation, a backstep controller is simulated for the end-effector trajectory tracking for a multicopter with a two-DoF robotic arm. In [18] an adaptive sliding mode controller is developed to the same aerial vehicle. In the experiment, the aerial manipulator takes off, flies toward an object whose position is known, grasps the object, and finally, as shown in Figure 6, releases it in another place. The sliding mode controller is proven to be robust to the environment interaction during manipulation.

In recent years, noticeable progress has been seen in the research area of aerial interaction with vertical surfaces. In [19] and [20], a small quadcopter was designed to avoid obstacles during free flight and perform physical interaction tasks. In Figure 7, the vehicle is writing the laboratory initials on a wall. A switching logic, based on the vehicle's velocity, is defined to identify if the quadcopter is free flying or interacting with the wall. In any case, the control law is based on model predictive control (MPC), a technique that computes a sub-optimal trajectory to minimize a cost function for a fixed amount of time while respecting state space and input constraints. For the free-flight case, the cost function includes terms to avoid obstacles, while during the physical interaction mode, hard constraints, like maximum attitude angle to prevent the vehicle from crashing into the wall, are implemented.

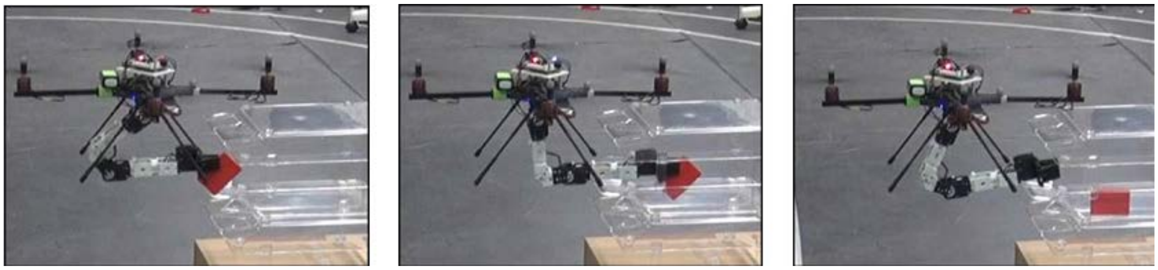


Figure 6. Quadrotor Releasing Block. Source: [18].

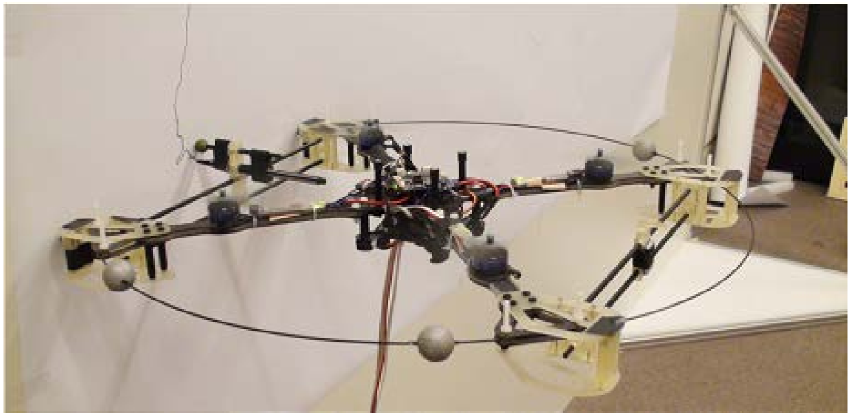


Figure 7. Quadrotor Writing on a Wall. Source: [19].

In [21], a tilt trirotor is employed to apply a larger force on a vertical surface, as in Figure 8. Since two of the propellers tilt while the fixed body remains in the horizontal plane, the aerial vehicle can produce a greater force on the object while keeping stability. The object is pushed by means of a flat end-effector with tactile sensors, for contact detection and switching mode capability. A gain-scheduled PID control is implemented, where the gains vary with the tilting angle of the two front rotors. The experimental validation demonstrated the aerial vehicle ability to successfully switch between modes, including free flight, collision, pushing statically, and pushing a moving object.

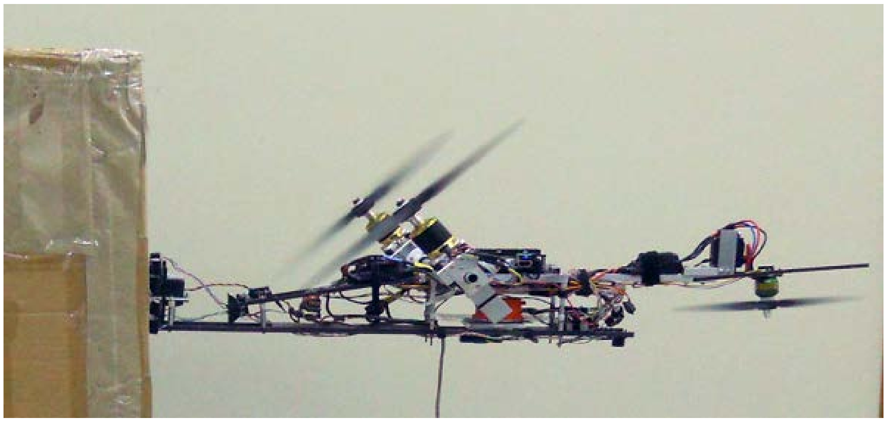


Figure 8. Trirotor Pushing a Vertical Object. Source: [21].

A quadrotor for tool tip force/torque operations is simulated in [22]. Initially, the dynamics of the aerial vehicle are derived in terms of the multicopter center of mass position, neglecting the mass of the tool fixed on the body. Then, the dynamic equations are written in terms of the tool tip position and decomposed in tangential and normal components to the wall surface. Because the multicopter is underactuated, the control law considers the inner dynamics of the system to achieve stability. The analysis shows that the system tends to be more stable during force torque operation if the tool is fixed above the center of mass of the aerial vehicle. To illustrate that, the paper states that a quadrotor with a tool tip fixed below the center of mass is comparable to an inverted pendulum.

The interaction control and collision reflexes for flying robots are developed based on estimated external wrench [23]. Two methods are presented to estimate the external forces, the momentum-based and the acceleration-based estimation. The first method is employed to estimate the torques due to the fact that the gyroscope sensors provide directly the angular rates. The external forces are estimated from the acceleration method because exteroceptive sensors are necessary to compute drift-free velocities estimations. The experiments tested four different kinds of reactions to wall collision: no reaction, stop, bounce, and impedance interaction. The results show that if no reaction is implemented, the vehicle crashes, and the smoothest response happens for the impedance control case.

In [24], a multicopter equipped with a manipulator is used to study the aerial manipulation on a vertical surface. The hybrid control switches from free-flight to interaction mode by means of a force sensor output threshold. During the contact with the wall, an impedance controller is responsible for the lateral and vertical movement (parallel to the wall), and a force controller is responsible for the longitudinal direction (normal to the wall). A similar experimental set up is described in [25] to validate a new impedance controller for physical interaction. The paper proposes a virtual dynamic system which is coupled to the real system by making the reference position of the virtual system be the actual position of the real system and inversely making the reference position of the real system be the actual position of the virtual system. The combined system has the external force to the virtual system as input and the external force of the real system as output. This new impedance controller is theoretically proven to be stable, which is also validated through simulations and experimental results. In [26], a novel free-flight controller was developed for the quadcopter with a prismatic joint. The movement of the prismatic joint induces attitude torques, changes the center of mass of the system and modifies the inertia matrix of the aerial vehicle. During free flight, when no manipulation is happening, the prismatic join is guided to improve the attitude control performance. Experiments and simulations show that the attitude error is significantly smaller for the controller that explores the manipulator dynamics. In [27], a passivity-based controller is proposed for a quadcopter equipped with a manipulator for aerial inspection. During free flight, the vehicle converges to the reference position while the interaction forces are null. After the contact with the vertical surface, the system works as a mass spring damped system, converging to a constant interaction force and steady state position error. The effectiveness of this impedance approach is proved through simulations and experiments, as shown in Figure 9, where the aerial vehicle smoothly docks to the wall. In [28], a variable impedance controller is designed for a quadcopter interacting with the environment. In a passivity-based controller, as in a mass spring system, the contact force is directly related to the position error and the stiffness of the system. So, an outer control layer is developed to change the stiffness constant in order to reach a desired interaction force.

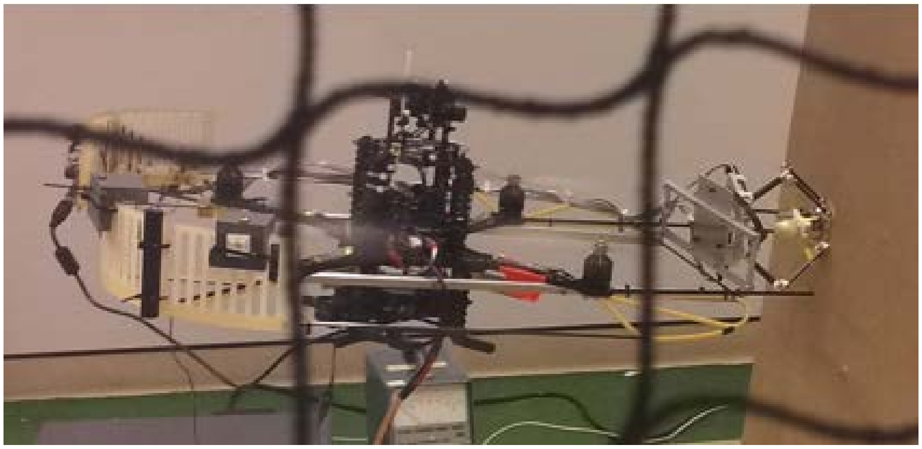


Figure 9. Aerial Interaction with a Vertical Surface. Source: [27].

A planar multicopter is known to be an underactuated system since the thrust force is limited to the vertical axis of the body-fixed frame. In [29], a hexacopter with canted propellers is proposed, so that each blade applies thrust with components in the vertical and in the horizontal plane. In Figure 10, this new configuration is shown, where the thrusters are bent. The hexacopter is then able to produce decoupled forces and torques with no restriction. In addition, it is shown experimentally to be capable of hovering with a tilt angle. With a manipulator attached to the bottom of the multicopter, the new structure allows better interaction with the environment.

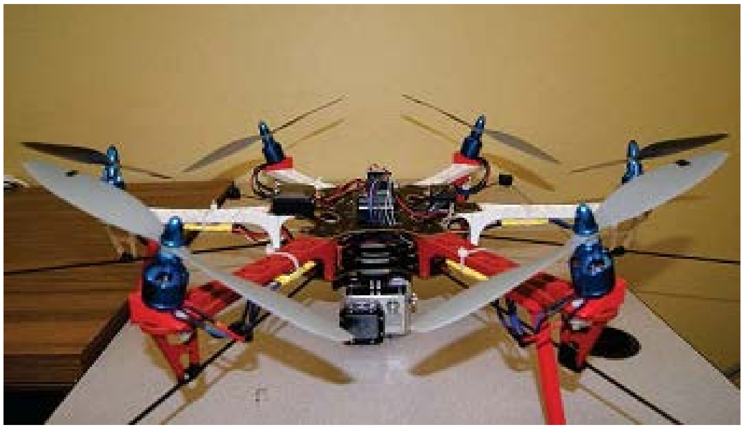


Figure 10. Hexacopter with Canted Propellers. Source: [29].

When a vehicle hovers close to a vertical surface, disturbance moments are induced on the body. In [30], the near-wall effect is investigated by means of computational tools and, by considering a single main rotor helicopter. Computed moments about *x* and *y* (rotor disk plane) are compared to those of other turbulence models from the literature to validate the proposed method. Finally, a disturbance observer is suggested to estimate the near wall moments and improve the performance of aerial vehicles flying close to a vertical surface.

The literature review shows that the aerial manipulation is an emerging area of study. The problem of grasping objects with unknown weight while moving a complex robotic arm with minimum impact on the attitude control has been successfully solved. There is also a considerable progress for multicopters interacting with vertical surfaces but equipped with simple moving structures, like prismatic joints or no robotic arm at all. Some of those vehicles are designed simply to push a wall or a moving object. Other vehicles are able to fly in contact with a vertical surface while moving laterally or vertically to write on a wall or to perform physical inspection; however, no experiment addresses the problem of a multicopter equipped with a robotic arm with many degrees of freedom interacting with a vertical surface. Such an aerial manipulator would be able to not only push the wall, but also to apply forces and/or torques in any direction.

# III. MATHEMATICAL MODEL

In this chapter, a mathematical model for a multicopter equipped with a robotic arm is developed. The equations representing the model are applied to the experimental aerial manipulator used in this research. Next, the near-wall effect is characterized and included in the model. Finally, a damped spring system is chosen to model the interaction force and torque on the wall, which are transformed by the end-effector Jacobian to be included in the model.

## A. MODEL OF A GENERIC AERIAL MANIPULATOR

A mathematical model for a generic aerial manipulator is developed in this section. In the following section, the model is applied to the Spacecraft Robotics Laboratory aerial vehicle, a hexacopter with a three-DoF robotic arm.

The dynamic equations are derived through the Euler-Lagrangian formulation. The torques and forces produced by the propellers, the torques produced by the revolute joint servos, the external interaction with the wall and the multicopter, and the robotic arm rigid body dynamics are considered. The propeller rigid body dynamics and any other aerodynamic effects are neglected since the proposed study comprises missions with no rapid-moving maneuvers.

The development of the dynamic equations follows the notations and conventions in [31]. Initially, the position and orientation of the multicopter and the robotic links are derived in terms of the generalized coordinates of the aerial manipulator. These equations are differentiated to obtain the velocities and angular rates in terms of Jacobian matrices and the first derivatives of the generalized coordinates. Next, the resultant expressions are applied to compute the kinetic and potential energy of the aerial manipulator. Finally, the energy expressions are applied in the Lagrangian equation of motion to derive the dynamic equations.

A multicopter equipped with an n-link robotic arm has $6 + n$ degrees of freedom. The vector $q = (x, y, z, \phi, \theta, \psi, \alpha_1, \dots, \alpha_n)^T$ denotes the generalized coordinates, where

$p_0 = (x, y, z)^T$ is the position of the multicopter center of mass expressed in the inertial frame, $\Phi = (\phi, \theta, \psi)^T$ are the Euler angles for the multicopter orientation, and $(\alpha_1, \dots, \alpha_n)^T$ are the robotic arm joints angles. The reference frames for the multicopter body and the robot links are illustrated in Figure 11. Let $p_{sk}$, where $k = 1, \dots, n$, be the absolute position of the $k^{th}$ robot joint expressed in the inertial frame, $p_k$ be the absolute position of the $k^{th}$ link's center of mass expressed in the inertial frame, and $\omega_k$ be its angular rate expressed in the respective link's body-fixed frame. The multicopter body-fixed frame is positioned according to the North East Down (NED) convention and the Euler angles are defined by the sequence of rotations yaw-pitch-roll, about the axes z, y, and x, respectively.

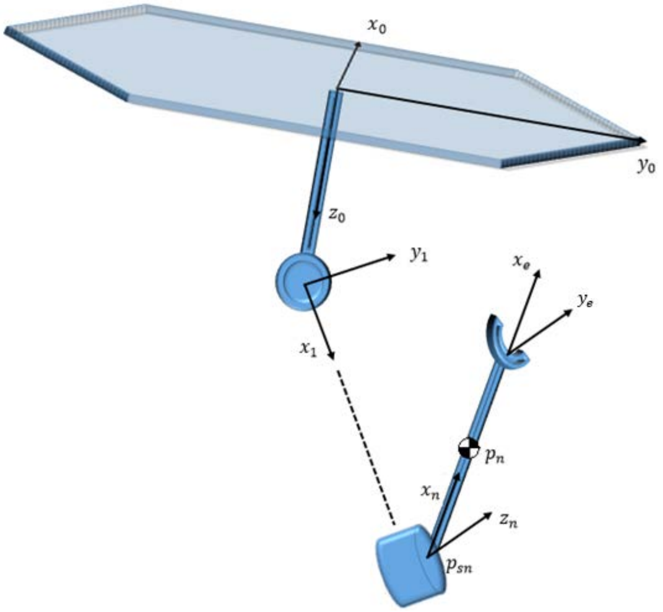


Figure 11. Aerial Manipulator Reference Frames

The rotation matrix denoting the orientation of the multicopter body-fixed frame with respect to the inertial frame is

$$R_{i0} = R_z(\psi) R_y(\theta) R_x(\phi) \quad (1)$$

where the three rotation matrices about the principal axes are given by

$$
\begin{aligned}
R_z(\psi) &= \begin{bmatrix} \cos(\psi) & \text{-}\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
R_y(\theta) &= \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ \text{-}\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \\
R_x(\phi) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \text{-}\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}.
\end{aligned}
\tag{2}
$$

Let $\omega_0$ be the multicopter's angular rate expressed in the body-fixed frame, which can be written in terms of the time derivatives of the Euler angles as [32]

$$\omega_0 = Q\dot{\Phi} \tag{3}$$

where

$$Q = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix}. \tag{4}$$

The linear velocity and angular rate of the multicopter body are combined into a six-dimensional vector denoted by

$$v_0 = \begin{bmatrix} \dot{p}_0 \\ \omega_0 \end{bmatrix}. \tag{5}$$

This vector is related to the time derivative of the generalized coordinates through a Jacobian matrix as follows:

$$v_0 = \begin{bmatrix} J_{p0} \\ J_{\omega 0} \end{bmatrix} \dot{q} = J_0 \dot{q} \tag{6}$$

where

$$J_0 = \begin{bmatrix} J_{p0} \\ J_{\omega 0} \end{bmatrix} = \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} & 0_{3\times n} \\ 0_{3\times 3} & Q & 0_{3\times n} \end{bmatrix}. \tag{7}$$

For each robotic arm link, the differential kinematics of the center of mass position, the orientation, and associated joint position can also be written in terms of Jacobian matrices

$$v_k = \begin{bmatrix} \dot{p}_k \\ \omega_k \end{bmatrix} = \begin{bmatrix} J_{pk} \\ J_{\omega k} \end{bmatrix} \dot{q} = J_k \dot{q} \tag{8}$$

and

$$\dot{p}_{sk} = J_{sk} \dot{q} \,. \tag{9}$$

The angular rate of each link is expressed recursively using the rotation matrix $R_{k,k-1}$, which denotes the orientation of the link *k-1* with respect to the link *k*. For *k=1*, $R_{10}$ is the rotation matrix denoting the orientation of the multicopter body with respect to the first link. The recursive angular rate equations are given by

$$\omega_k = R_{k,k-1}\omega_{k-1} + \begin{bmatrix} 0 & 0 & \dot{\alpha}_k \end{bmatrix}^T \tag{10}$$

$$J_{\omega k} = R_{k,k-1} J_{\omega,k-1} + \begin{bmatrix} & 0 & \\ 0_{3\times(6+k-1)} & 0 & 0_{3\times(n-k)} \\ & 1 & \end{bmatrix} \tag{11}$$

where $0_{i\times j}$ is a null matrix with *i* rows and *j* columns. These null matrices are concatenated with the vector $[0 \quad 0 \quad 1]^T$ to produce a matrix with zeros, except being equal to one for the element at the third row and the *(k+6)*$^{th}$ column. The end-effector orientation is defined relative to the orientation of the last link transformed by a constant rotation matrix $R_{en}$ as follows:

$$J_{\omega e} = R_{en} J_{\omega n} \,. \tag{12}$$

Let the parameter $p_{01}$ be the displacement of the first joint relative to the center of mass of the multicopter expressed in the multicopter body-fixed frame. So, the kinematics of the first joint is given by

$$p_{s1} = p_0 + R_{i0} p_{01} \tag{13}$$

$$\dot{p}_{s1} = \dot{p}_0 + R_{i0} \omega_0^{\times} p_{01} \tag{14}$$

$$J_{s1} = J_{p0} - R_{i0} p_{01}^{\times} J_{\omega 0} \ . \tag{15}$$

The body-fixed reference frame $\begin{bmatrix} \hat{x}_k & \hat{y}_k & \hat{z}_k \end{bmatrix}$ is placed at the *k^th^* joint, with $x_k$ parallel to the link and $z_k$ parallel to the axis of rotation of the *k^th^* joint, as in Figure 11. Let $R_{ik}$ be the rotation matrix denoting the orientation of the body-fixed frame with respect to the inertial frame. Then, the position and velocity of the joints are computed recursively by means of

$$p_{s,k+1} = p_{sk} + R_{ik} \begin{bmatrix} l_k & 0 & 0 \end{bmatrix}^T \tag{16}$$

$$\dot{p}_{s,k+1} = \dot{p}_{sk} + R_{ik} \omega_k^{\times} \begin{bmatrix} l_k & 0 & 0 \end{bmatrix}^T \tag{17}$$

$$J_{s,k+1} = J_{sk} - R_{ik} \begin{bmatrix} l_k \\ 0 \\ 0 \end{bmatrix}^{\times} J_{\omega k} \ . \tag{18}$$

The position and velocity of each link at the center of mass are also expressed recursively as follows:

$$p_k = p_{sk} + R_{ik} \begin{bmatrix} l_{ck} & 0 & 0 \end{bmatrix}^T \tag{19}$$

$$\dot{p}_k = \dot{p}_{sk} + R_{ik} \omega_k^{\times} \begin{bmatrix} l_{ck} & 0 & 0 \end{bmatrix}^T \tag{20}$$

$$J_{pk} = J_{sk} - R_{ik} \begin{bmatrix} l_{ck} \\ 0 \\ 0 \end{bmatrix}^{\times} J_{\omega k} \ . \tag{21}$$

Similarly, the end-effector Jacobian is given by

$$J_{pe} = J_{sn} - R_{in} \begin{bmatrix} l_e \\ 0 \\ 0 \end{bmatrix}^{\times} J_{\omega n} \, . \tag{22}$$

Let $I_k$ and $m_k$ be the moment of inertia matrix and the mass of the link *k*. The kinetic energy of the multicopter (*k=0*) and the links (*k = 1, ... , n*) are defined by

$$K_k = \frac{1}{2} v_k^T \begin{bmatrix} m_k I_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & I_k \end{bmatrix} v_k = \frac{1}{2} v_k^T M_k v_k . \tag{23}$$

From Equations (8) and (23), the expressions for the kinetic energy of each link and for the total kinetic energy are given by

$$K_k = \frac{1}{2} \dot{q}^T J_k^T M_k J_k \dot{q} = \frac{1}{2} \dot{q}^T B_k \dot{q} \tag{24}$$

$$K = \frac{1}{2} \dot{q}^T \left( \sum_{k=0}^{n} B_k \right) \dot{q} = \frac{1}{2} \dot{q}^T B \dot{q} \ . \tag{25}$$

The potential energy depends on the masses and the vertical component of the center of mass position of each link as follows:

$$P_k = \begin{bmatrix} 0 & 0 & -m_k . g \end{bmatrix} . p_k \tag{26}$$

$$P = \sum_{k=0}^{n} P_k \ . \tag{27}$$

Substituting the kinetic and the potential energies in the Lagrange equation, we have the motion equation for each degree of freedom

$$\frac{d}{dt}\left( \frac{\partial K}{\partial \dot{q}_i} \right) - \frac{\partial K}{\partial q_i} + \frac{\partial P}{\partial q_i} = \tau_i \tag{28}$$

where $i = 1, ..., n+6$ .

Combining Equations (25), (27), and (28) yields the dynamic equations of the system

$$\frac{d}{dt}\left( B(q)\dot{q} \right) - 0.5 \begin{bmatrix} \dot{q}^T \dfrac{\partial B}{\partial q_1} \dot{q} \\ \vdots \\ \dot{q}^T \dfrac{\partial B}{\partial q_{n+6}} \dot{q} \end{bmatrix} + \begin{bmatrix} \dfrac{\partial P}{\partial q_1} \\ \vdots \\ \dfrac{\partial P}{\partial q_{n+6}} \end{bmatrix} = \tau + \tau_{ext} \tag{29}$$

and

$$B(q)\ddot{q}+\dot{B}(q)\dot{q}-0.5\begin{bmatrix}\dot{q}^T\dfrac{\partial B}{\partial q_1}\\ \vdots \\ \dot{q}^T\dfrac{\partial B}{\partial q_{n+6}}\end{bmatrix}\dot{q}+\begin{bmatrix}\dfrac{\partial P}{\partial q_1}\\ \vdots \\ \dfrac{\partial P}{\partial q_{n+6}}\end{bmatrix}=\tau+\tau_{ext} \tag{30}$$

where $\tau$ is the generalized input force vector and $\tau_{ext}$ is the generalized external forces vector at the joint level.

Denoting the partial derivative matrices by *N* and *G* allows Equation (30) to be written in a compact manner as

$$B(q)\ddot{q}+\dot{B}(q)\dot{q}-\frac{1}{2}N(q,\dot{q})\dot{q}+G(q)=\tau+\tau_{ext}\,. \tag{31}$$

This equation is further simplified by introducing a matrix $C(q,\dot{q})$, which is related to the Coriolis and centrifugal effects, unifying the terms that depend on $\dot{q}$, as follows:

$$B(q)\ddot{q}+C\left(q,\dot{q}\right)\dot{q}+G(q)=\tau+\tau_{ext}\,. \tag{32}$$

Although the immediate choice for *C* is, from Equation (31), $C=\dot{B}-0.5N$, the solution for this matrix is not unique [31]. In other words, there are multiple solutions for *C* for which the column vector $C\dot{q}$ is the same. Most of the authors in robotics choose $C=0.5\dot{B}+0.5N^T-0.5N$, which is also proven to be a valid solution. Rearranging the last equation results in $\dot{B}-2C=N-N^T$, which is a skew-symmetric term. Some stability analysis based on energy functions and *Lyapunov* theory uses this convenient skew-symmetry property to prove stability. In this study, however, the matrix *C* is chosen in order to reduce the computational load in simulations. By noting the elements $b_{ij}$ and $n_{ij}$ of the matrices *B* and *N*, respectively, the Coriolis and centrifugal terms in the motion equation are expressed as follows:

$$N\dot{q}=\begin{bmatrix}\vdots\\ \displaystyle\sum_{j=1}^{n+6}\sum_{k=1}^{n+6}\frac{\partial b_{jk}}{\partial q_i}\dot{q}_k\dot{q}_j\\ \vdots\end{bmatrix}=\begin{bmatrix} & \vdots & \\ \cdots & n_{ij}=\displaystyle\sum_{k=1}^{n+6}\frac{\partial b_{jk}}{\partial q_i}\dot{q}_k & \cdots\\ & \vdots & \end{bmatrix}\dot{q} \tag{33}$$

and

$$\dot{B}\dot{q} = \begin{bmatrix} \vdots \\ \sum_{j=1}^{n+6} \sum_{k=1}^{n+6} \frac{\partial b_{ij}}{\partial q_k} \dot{q}_k \dot{q}_j \\ \vdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & \dot{b}_{ij} = \sum_{k=1}^{n+6} \frac{\partial b_{ij}}{\partial q_k} \dot{q}_k & \cdots \\ & \vdots & \end{bmatrix} \dot{q}\,. \tag{34}$$

An interesting relation between these two equations is found if the matrix $N$ is transposed in Equation (33), which leads to

$$N^T \dot{q} = \begin{bmatrix} \vdots \\ \sum_{j=1}^{n+6} \sum_{k=1}^{n+6} \frac{\partial b_{ik}}{\partial q_j} \dot{q}_k \dot{q}_j \\ \vdots \end{bmatrix} = \begin{bmatrix} & \vdots & \\ \cdots & n_{ji} = \sum_{k=1}^{n+6} \frac{\partial b_{ik}}{\partial q_j} \dot{q}_k & \cdots \\ & \vdots & \end{bmatrix} \dot{q}\,. \tag{35}$$

If the summations in Equation (34) are switched, the result is identical to the summations in Equation (35)

$$\sum_{j=1}^{n+6} \sum_{k=1}^{n+6} \frac{\partial b_{ij}}{\partial q_k} \dot{q}_k \dot{q}_j = \sum_{j=1}^{n+6} \sum_{k=1}^{n+6} \frac{\partial b_{ik}}{\partial q_j} \dot{q}_k \dot{q}_j \tag{36}$$

which implies

$$N^T \dot{q} = \dot{B}\dot{q}\,. \tag{37}$$

This result allows to determine the dynamic system without the need to compute the matrix $\dot{B}$, either analytically or by numerically differentiating the matrix B. Then, in the simulations and controllers presented in this study, once the matrix $N$ is computed, the matrix $C$ is defined as

$$C(q,\dot{q}) = N(q,\dot{q})^T - 0.5N(q,\dot{q})\,. \tag{38}$$

When the manipulator end-effector interacts with a wall, torques and forces are induced on the multicopter center of mass and torques are induced on the joints. The external interaction forces and torques exerted by the environment on the end-effector are mapped to the generalized coordinate space as follows [31]:

$$\tau_{ext} = \begin{bmatrix} J_{pe} \\ J_{\omega e} \end{bmatrix}^T \begin{bmatrix} F_e \\ \tau_e \end{bmatrix} = J_e^T \begin{bmatrix} F_e \\ \tau_e \end{bmatrix}. \tag{39}$$

The input forces and torques are produced by the rotors of the multicopter and the servos in the robotic arm joints. They are mapped to the generalized coordinate level as follows [1]:

$$\tau = \begin{bmatrix} F_0 \\ \tau_\Phi \\ \tau_\alpha \end{bmatrix} = \begin{bmatrix} R_{i0} & 0 & 0 \\ 0 & Q^T & 0 \\ 0 & 0 & I_{n\times n} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 0 & 0 & -F_t \end{bmatrix}^T \\ \tau_p \\ \tau_\alpha \end{bmatrix}. \tag{40}$$

When a single blade rotates, it produces a torque and a force along the vertical axis (orthogonal to the plane of rotation). In a multicopter with several propellers, those forces and torques are combined to result in a force and a torque on the center of mass of the multicopter body. In Equation (40) the term $F_t$ is the vertical force in the multicopter body-fixed frame produced by the thrusters, and $\tau_p$ is the resultant torque. These terms are mapped to the generalized coordinate system resulting in $F_0$, the force expressed in terms of the inertial frame, and $\tau_\Phi$, the torque associated to the multicopter Euler angles. Finally, the vector $\tau_\alpha$ represents the torques applied by the servos on the robotic arm.

## B. HEXACOPTER WITH A THREE-DOF ROBOTIC ARM

In this section, the motion equations developed for multicopters with n-DoF manipulators in the previous section are applied to a hexacopter with a 3-DoF robotic arm. This hexacopter with a three-DoF robotic arm is available in the Spacecraft Robotics Laboratory. It is used in the subsequent simulation study and experimentation. In Figure 12, the orientation of the joints are shown. The drawing illustrates the link coordinate systems at the initial configuration for the angles $\alpha_1$, $\alpha_2$, and $\alpha_3$ which are all zero with $x_1$ , $x_2$ , and $x_3$ pointing in the same (downward) direction.

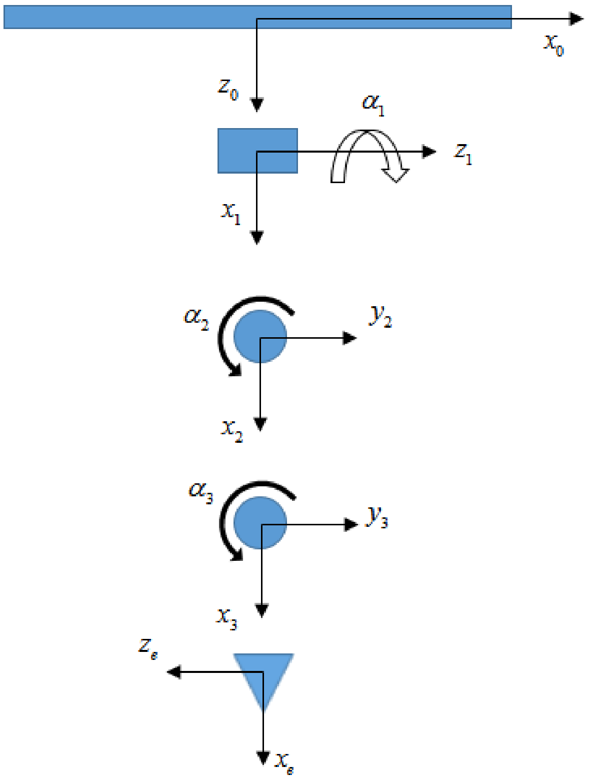


Figure 12. Join Orientation of the Robotic Arm

With the link coordinate systems as defined, the link rotation matrices are given by

$$R_{01}(\alpha_1)=\begin{bmatrix} 0 & 0 & 1 \\ -\sin(\alpha_1) & -\cos(\alpha_1) & 0 \\ \cos(\alpha_1) & -\sin(\alpha_1) & 0 \end{bmatrix} \tag{41}$$

$$R_{12}(\alpha_2)=\begin{bmatrix} \cos(\alpha_2) & -\sin(\alpha_2) & 0 \\ 0 & 0 & -1 \\ \sin(\alpha 2) & \cos(\alpha_2) & 0 \end{bmatrix} \tag{42}$$

$$R_{23}(\alpha_3)=\begin{bmatrix} \cos(\alpha_3) & -\sin(\alpha_3) & 0 \\ \sin(\alpha_3) & \cos(\alpha_3) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{43}$$

The end-effector-fixed frame is also attached to the third link as in Figure 12. As a result, the rotation matrix between this frame and the third link coordinate system is constant and given by

$$R_{3e} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{44}$$

The force and torque induced on the center of mass of the multicopter depend on the number of thrusters, their geometry and which of the propellers rotate clockwise and counterclockwise. In Figure 13, the hexagonal geometry for the hexacopter is shown, as well as the number that identifies each blade and their direction of rotation.

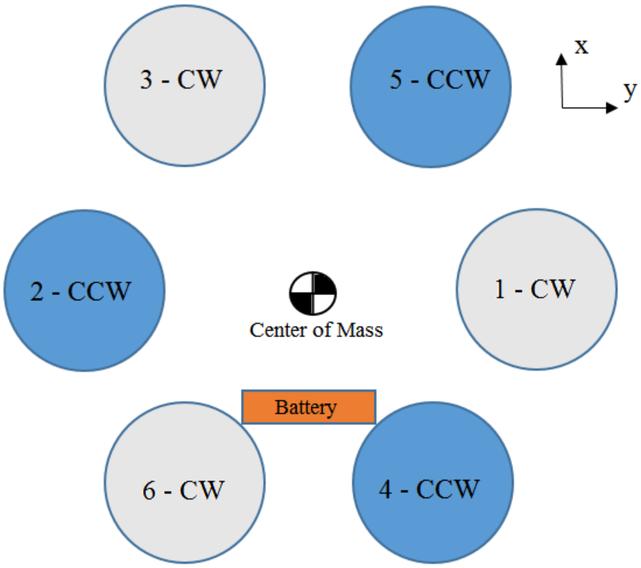


Figure 13. Top View of the Hexacopter

Given the PWM (pulse width modulation) input to each motor, the resultant forces and torques are calculated from the expressions [33]

$$
\begin{aligned}
F_t &= K_F \sum\nolimits_{k=1}^{6} \left( PWM_k - PWM_0 \right) \\
\tau_x &= \frac{l_H K_F}{2} \left( PWM_2 + PWM_3 + 2 \times PWM_6 - 2 \times PWM_1 - PWM_4 - PWM_5 \right) \\
\tau_y &= \frac{\sqrt{3} l_H K_F}{2} \left( PWM_3 + PWM_5 - PWM_6 - PWM_4 \right) \\
\tau_z &= K_\tau \left( PWM_2 + PWM_4 + PWM_5 - PWM_1 - PWM_3 - PWM_6 \right) \\
\tau_p &= \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T
\end{aligned}
\tag{45}
$$

where $K_F$, $PWM_0$, and $K_\tau$ are the propulsion system constants, $l_H$ is the distance between the centers of adjacent propellers, and $PWM_k$ is the input to the motor $k$.

As shown in Figure 13, the battery of the hexacopter is modeled as a separate rigid body. Consequently, the Lagrange equations are changed to include the battery dynamics. If the battery position with respect to the hexacopter, expressed in terms the body-fixed frame, is denoted by $p_{0b}$, then its velocity and *Jacobian* are

$$\dot{p}_b = \dot{p}_0 + R_{i0} \omega_0^\times p_{0b} \tag{46}$$

$$J_b = J_{p0} - R_{i0} p_{0b}^\times J_{\omega 0} \, . \tag{47}$$

From the kinematics of the battery, the kinetic and potential energy are computed by

$$K_b = \frac{1}{2} \dot{p}_b^T m_b I_{3\times 3} \dot{p}_b = \frac{1}{2} \dot{p}_b^T M_b \dot{p}_b \tag{48}$$

$$K_b = \frac{1}{2} \dot{q}^T J_b^T M_b J_b \dot{q} = \frac{1}{2} \dot{q}^T B_b \dot{q} \tag{49}$$

$$P_b = \begin{bmatrix} 0 & 0 & -m_b.g \end{bmatrix} . \left( p_0 + R_{i0} p_{0b} \right) = \begin{bmatrix} 0 & 0 & -m_b.g \end{bmatrix} . p_b \, . \tag{50}$$

where $m_b$ is the mass of the battery and its moment of inertia is approximated to be zero, since its dimensions are small, resulting in a compact body. Then, $K_b$ and $P_b$ are added to the summations in Equation (25) and Equation (27), respectively, to include the battery dynamics to the model.

## C. NEAR-WALL EFFECT

When the aerial manipulator approaches a wall, the presence of this vertical obstacle acts on the air flow and consequently changes the aerodynamic forces. Torques on the body result in undesirable movement, affecting the flight accuracy and safety. In order to better approach this problem, the near-wall effect is modeled. Initially, the wall disturbance on a single-blade helicopter is addressed and the results are expanded to the multicopter problem.

When a single main rotor vehicle flies far from any obstacle, the airflow is theoretically symmetric with respect to the centered vertical axis. However, if the helicopter approaches a vertical surface, as shown in Figure 14, the airflow becomes asymmetric. In [30], a detailed computational study shows that the near-wall disturbance results in two relevant components of torque, about axes *x* and *y*, as shown in Figure 15. The closer the helicopter is from the surface, the larger the near-wall effect is expected to be.

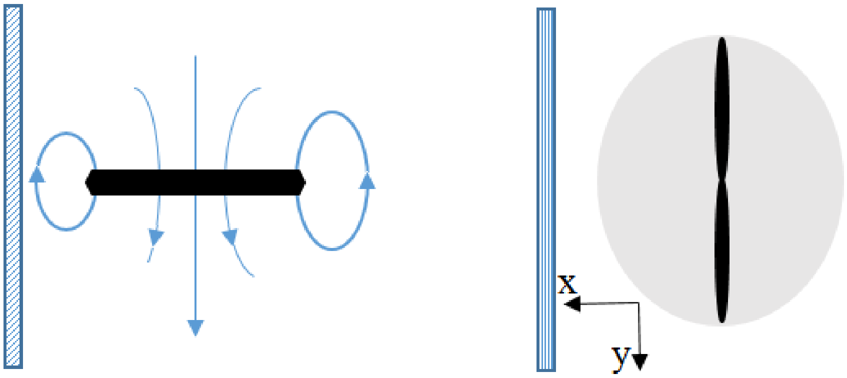
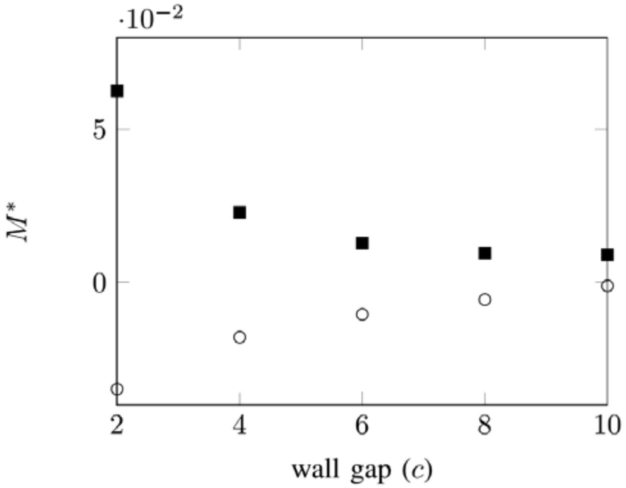


Figure 14. Side and Top Views of a Helicopter Near a Vertical Surface

The near-wall disturbance model for a single blade is taken as reference to study the case of a hexacopter hovering close to a vertical wall. In this study, the hexacopter orientation is aligned to the wall as in Figure 16. The three blades rotating clockwise and the other three rotating counterclockwise are positioned symmetrically. The single blade study implies that, for this symmetric configuration, the near-wall torques about the *x* axis are cancelled, while the torques about the *y* axis are added. For that reason, the near-wall effect for a hexacopter approaching perpendicularly an infinite wall is modeled as a torque about the *y* axis (nose down pitch movement), inversely related with the distance from the wall. Experiments are described in Chapter VII to measure the torque acting on the hexacopter body for different distances. A suitable curve is fitted to experimental data to determine an adequate mathematical expression relating the torque and the distance. In Chapter V, a modification is proposed to the controller to cancel the near-wall torque.



Square markers are moments about y axis, and circle markers are moments about x axis. The wall gap is given in terms of the chord length of the helicopter blade. The moments are non-dimensionalised.

Figure 15. Near-Wall Moment for a Single Blade Helicopter. Source: [30].

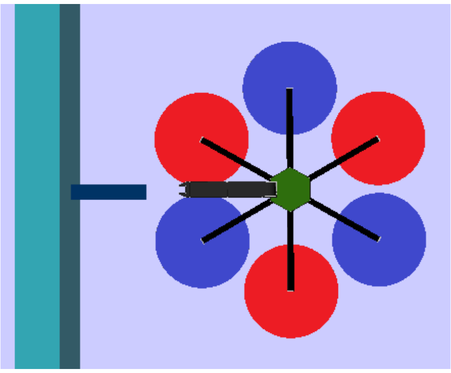


Figure 16. Aerial Manipulator Hovering Near a Vertical Wall

## D. SIMULATION MODEL OF INTERACTION FORCE AND TORQUE

When an aerial manipulator performs a task to interact with a wall as shown in Figure 17, the end-effector exerts a force $F_e$ and torque $\tau_e$ on the wall. The force and torque are modeled a damped spring system as follows:

$$\begin{bmatrix} F_e \\ \tau_e \end{bmatrix} = K_{pw} \begin{bmatrix} \Delta p_e \\ \Delta \Phi_e \end{bmatrix} - K_{dw} \begin{bmatrix} \dot{p}_e \\ \omega_e \end{bmatrix} . \tag{51}$$

The interaction forces and torques constrain the movement of the end-effector as if it was holding a rigid object on the wall. By choosing high gain values, a rigid system is addressed, while for smaller values, the flexibility of the robotic arm and the manipulated object, which are not ideally rigid, are simulated.

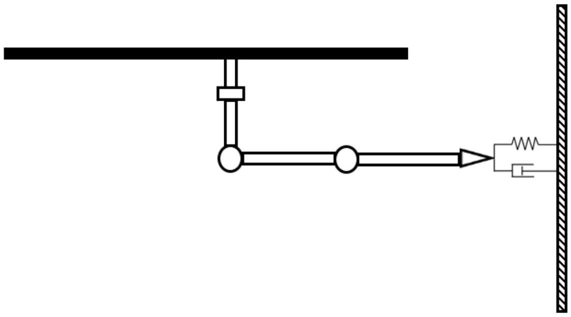


Figure 17. Aerial Manipulator Interacting with a Wall

THIS PAGE INTENTIONALLY LEFT BLANK

# IV. SYSTEM IDENTIFICATION

An aerial manipulator is a complex dynamic system, with multiple degrees of freedom. The operation comprised by approaching and interacting with a vertical wall is even more difficult, because of the air disturbance from near-wall effect and the risk of collision with the wall. It is challenging to design a controller robust not only to the disturbances, but also to modeling errors. That is why it is important to obtain a precise model, which depends on a good system identification to determine the parameters accurately. In addition, a reasonable dynamic model is also the basis for the simulation model, which must be comparable to the real system. In this chapter the methodologies to measure the inertial and electrical parameters of the hexacopter and the robotic arm are described.

## A. HEXACOPTER

In [33], the authors describe a methodology to determine the inertial and thruster parameters for multicopters and part of this procedure is applied to measure the main parameters of the hexacopter employed in this research.

Initially, with a ruler and a caliper, the side of the hexagon and the relative position of the battery with respect to the center of the hexagon are measured. Then, the masses of the hexacopter and the battery are measured separately with a scale.

The moments of inertia are measured by suspending the hexacopter in a pendulum setup as in Figure 18. Three experiments are executed, one for each body-fixed frame axis. In addition, the tests are carefully configured to produce small angles. As a result, the pendulum dynamic equation $\left(m_0 l_0 + m_{rod} l_{rod} + I + I_{rod}\right)\ddot{\theta} + \left(m_0 l_0 + m_{rod} l_{rod}\right) g\theta = 0$ is equivalent to the simple harmonic motion equation $\ddot{\theta} + \left(2\pi / T\right)^2 \theta = 0$, resulting in

$$I = \frac{T^2}{4\pi^2}\left(m_0 l_0 + m_{rod} l_{rod}\right) g - m_0 l_0^2 - m_{rod} l_{rod}^2 - I_{rod} \tag{52}$$

where $T$ is the period of oscillation, $m_0$ is the mass of the hexacopter, $m_{rod}$ is the mass of the rod, $l_0$ is the distance from the hexacopter center of mass to the pivot, $l_{rod}$ is distance from the rod center of mass to the pivot, and $I_{rod}$ is the moment of inertia of the rod. In Equation (52), all parameters except $T$ are directly measurable. A motion capture system is used to record the angle of oscillation over the time and consequently, measure accurately the period $T$.

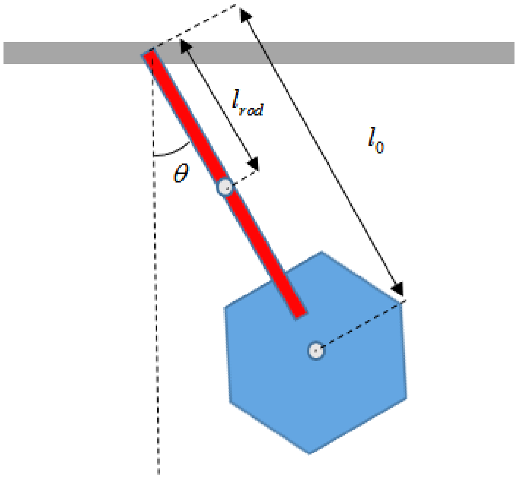


Figure 18. Measurement Setup for the Moment of Inertia about z (Yaw)

The constants $K_F$ and $PWM_0$ from Equation (45) are determined from a hovering experiment, like in Figure 19. The relationship that relates the thrust force $F_t$ and the PWM input is approximated to a first-order equation. Consequently, with two points the equation of the line, in terms of $K_F$ and $PWM_0$, is determined. The first point corresponds to $F_t=0$, and is determined by the $PWM = PWM_0$, where the propellers start to rotate. The second point is determined by the hovering condition, when the total trust and the weight of the vehicle cancel each other. Let $PWM_g$ be the input that, applied to all rotors, would produce a force $F_t$ with the same magnitude of the weight of the multicopter, as follows:

$$F_t = m_0 g = 6 \times K_F \left( PWM_g - PWM_0 \right). \tag{53}$$

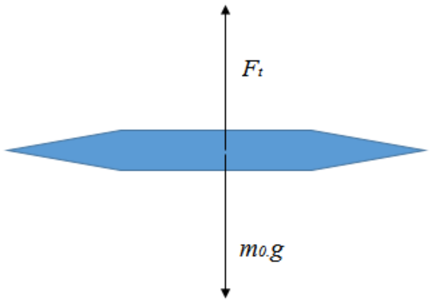


Figure 19. Hovering Setup for Thrust Measurement

In the Spacecraft Robotics Laboratory there is a granite testbed for frictionless experiments. The robots blow compressed air creating a thin layer of air between the granite and the robots. This testbed is in the same laboratory where the aerial manipulator experiments are done, with the motion capture system. The constant $K_\tau$ from Equation (45) is measured by placing the hexacopter on the top of a floating robot, as in Figure 20, and rotating only the three clockwise rotating blades. Since the same PWM input is sent to the three propellers and they rotate to the same direction, the torque produced about the vertical axis is equivalent to three times the torque applied by each blade. Some tests are run with different levels of PWM input and the resultant angular acceleration in each case is measured with the motion capture system. The inertial characteristics of the robot and the hexacopter are previously measured, which allows to compute the torque for each PWM input. Finally, the first order curve that relates the PWM input to the propeller torque is fitted to the set of measurements.

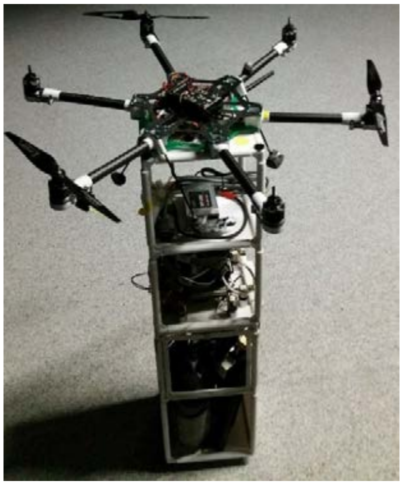


Figure 20. Floating Robot Setup for Torque Measurement

Applying the aforementioned methodology, the parameters of the experimental hexacopter were obtained and are presented in Table 1.

Table 1. Hexacopter Parameters

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $p_{0b}$ (m) | $\begin{bmatrix} -0.0635 \\ 0 \\ 0.0540 \end{bmatrix}$ | $I_0$ (Kg.m$^2$) | $\begin{bmatrix} 0.0266 & 0 & 0 \\ 0 & 0.0266 & 0 \\ 0 & 0 & 0.0508 \end{bmatrix}$ |
| $l_h$ (m) | 0.2750 | $PWM_0$ (μs) | 1100 |
| $m_0$ (Kg) | 1.1690 | $K_F$ ( N/μs ) | 0.0061 |
| $m_b$ (Kg) | 0.2740 | $K_\tau$ ( N.m/μs ) | 0.0001 |

## B. ROBOTIC ARM

In this section, the identification of the inertial and servos parameters of the robotic arm is detailed. Initially, the constant position $p_{01}$ of the first joint with respect to the hexacopter body-fixed frame was measured. Later, the robotic arm was disassembled to measure the properties of each link separately. The end-effector was considered as part of the last link.

For each link, the mass, the length, the center of mass position, and the moment of inertia were obtained. The mass and the length were measured simply by using a scale and a ruler. The static equilibrium illustrated in Figure 21 provides the mathematical equation $l_{cm}.mg = F_{sc}l$, where $F_{sc}$ is the force read from the scale, $l$ is the length of the link, and $m$ is its mass. So, the center of mass $l_{cm}$ is obtained by

$$l_{cm} = \frac{F_{sc}}{mg}l \ . \tag{54}$$

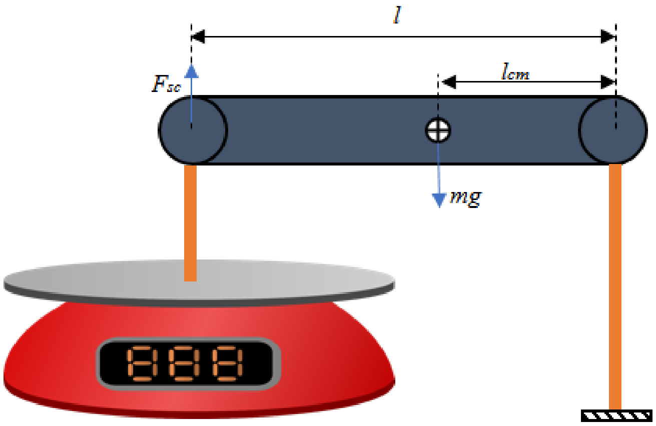


Figure 21. Measurement of the Center of Mass of a Link

The moment of inertia of each link was measured by making pendulum oscillations similarly to the hexacopter experiment, but without a supporting rod. In addition, the links are approximated to a symmetric long bar. So, the moment about *x* in the link-fixed frame is approximated to zero and the moments of inertia about y and z are equal, and computed as follows:

$$I_{y,z} = \frac{T^2}{4\pi^2}ml_{cm}g - ml_{cm}^2 . \tag{55}$$

The mathematical model derived for the aerial manipulator assumes that the input to the robotic arm is the torque on the servos. However, the robotic arm servos come from factory with an internal controller that rotates the joint to a goal position at a user specified angular velocity. The servos give position, velocity and torque feedback, but the torque feedback is not accurate and cannot be used in practice. The internal controller is not known, but the experiments show that the performance of the servos is excellent. In Figure 22, the feedback from one of the servos, during a flight experiment, is shown, when the joint is rotating from $\pi / 2$ radians to zero. These almost perfect straight lines segments indicate that, although the controller structure is not known, the system behavior is simple and predictable, and the simulations and controllers can be designed in terms of the robotic arms dynamics and joints kinematics. The servos were configured to rotate at a slow constant velocity. Consequently, the relevant parameter for the servo is the angular velocity of reference, here denoted by $\omega_{servo}$, which was determined by measuring the slope of the curve.

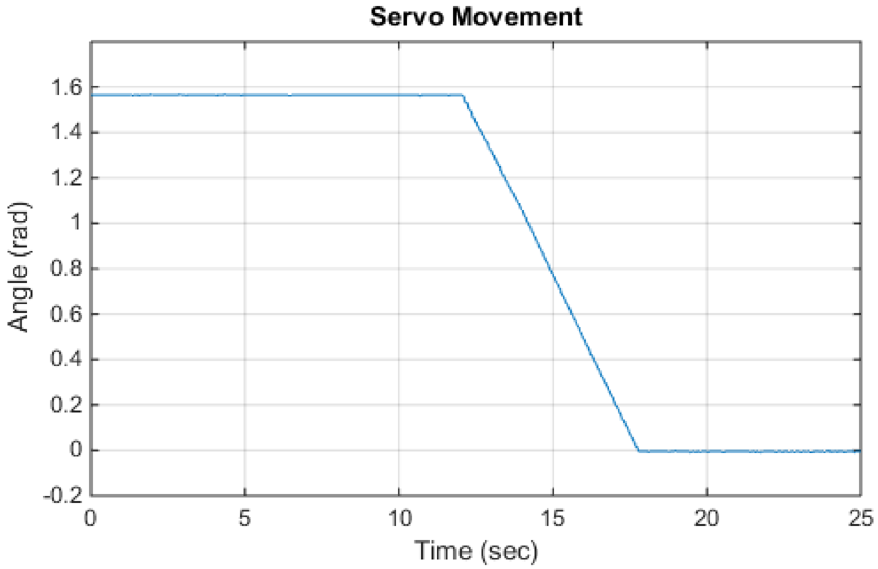


Figure 22. Servo Feedback during a Flight Experiment

The inertial and servos parameters of the experimental robotic arm with three links are presented in Table 2.

Table 2. Robotic Arm Parameters

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| $p_{01}$ (m) | $\begin{bmatrix}0.0540 & 0 & 0.0730\end{bmatrix}^T$ | $l_{cm1}$ (m) | 0.0438 |
| $l_1$ (m) | 0.0698 | $l_{cm2}$ (m) | 0.0735 |
| $l_2$ (m) | 0.1064 | $l_{cm3}$ (m) | 0.0703 |
| $l_3$ (m) | 0.1175 | $I_{y,z1}$ (Kg.m$^2$) | 6.024 x 10–$^5$ |
| $m_1$ (Kg) | 0.075 | $I_{y,z2}$ (Kg.m$^2$) | 1.1395 x 10–$^4$ |
| $m_2$ (Kg) | 0.097 | $I_{y,z3}$ (Kg.m$^2$) | 1.3760 x 10–$^4$ |
| $m_3$ (Kg) | 0.102 | $\omega_{servo}$ (rad/sec) | 0.2746 |

THIS PAGE INTENTIONALLY LEFT BLANK

# V. CONTROLLER DESIGN

Studies in aerial manipulation have already accomplished the control of the end-effector position during free flight and while the robot is picking or placing objects. In this research, a new problem is explored, in which an aerial manipulator is used to perform manipulation tasks on a vertical wall. In order to make progress in this area, two particular issues are addressed, the control of an aerial manipulator under near-wall disturbances, and the control of an aerial vehicle while applying forces and torques on a vertical surface. In order to investigate the near-wall effect, an aerial manipulator controller for free flight based on the current state of the art is first implemented. This controller is described in details, with the some modifications to adjust the control solution to the experimental setup. Once the free-flight controller is implemented, the near-wall effect can be characterized experimentally, and the modification to the controller is proposed to compensate the near-wall effect. Finally, for the torque/force interaction on the wall, a new controller is developed.

## A. FREE-FLIGHT CONTROL

The main issue in aerial manipulation free flight is how to control the robotic arm and the multicopter, given that their dynamics are coupled. For this reason, the controller presented here, which is a result of the recent progress in this area, is based on the dynamic model of the system as a whole [1]. Consequently, the dynamics of the robotic arm is taken into consideration in the attitude controller design. Previous research has shown that this controller has good performance in terms of end-effector trajectory accuracy [13]. In this section, a brief introduction on multicopter control is presented, in order to provide background for a better understanding of an aerial manipulator free-flight controller. Later, the aerial manipulator free-flight controller is detailed, as well as how it is changed to approach the near-wall effect with the hexacopter and robotic arm used in this research.

In a multicopter, as shown in Equation (45), by managing the propellers PWM inputs, the controller applies a thrust force in the vertical direction, in the body-fixed frame, and a torque in three dimensions, resulting in four decoupled inputs. That is why multicopters are known to be underactuated systems, since they control directly only four of its six degrees of freedom. In other words, if the multicopter is hovering, the force vector does not have horizontal components, and a lateral motion is only achieved by tilting the multicopter to project a thrust component on the horizontal plane to the desired direction. A popular solution for the underactuation is to design the controller in two layers, as presented in [34], [4], and [35]. In Figure 23, the idea of controlling the position by rolling and pitching the multicopter is illustrated.

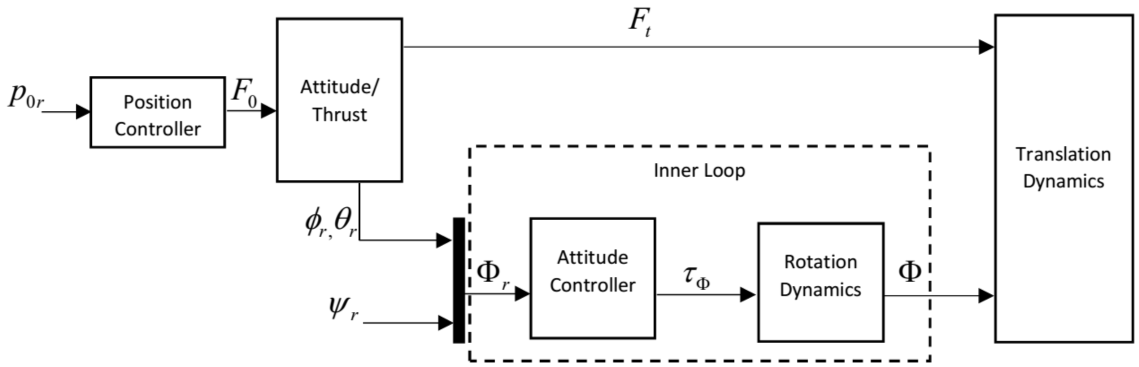


Figure 23. Multicopter Free-Flight Controller

In this case, the desired roll angle $\phi_r$ and pitch angle $\theta_r$ are defined as auxiliary inputs to the controller inner loop, and the desired condition or way point is defined in terms of a position vector $p_{0r}$ and a yaw angle $\psi_r$. Based on the desired translational movement, a position controller outputs the total force to be acting on the multicopter, which gives the force $F_0$, in the inertial frame, applied by the thrusters. The block *Attitude/Thrust* computes the thrust force $F_t$ and the attitude angles $\phi_r$ and $\theta_r$. The attitude reference is connected to the inner-loop block, where the attitude controller outputs the torques on the body-fixed frame so that the attitude $\Phi$ converges to the reference $\Phi_r$. For this two-layer controller, a constraint is that the attitude controller is faster than the outer-loop controller. With this constraint, from the position controller point of view, $\Phi \cong \Phi_r$ and the force that is actually applied on the multicopter by the thrusters is approximately $F_{0.}$ Additionally, the design process is very simple, since the two controllers can be designed separately. The outer-loop controller is designed based on the translational dynamics, and the inner-loop controller is designed based on the rotational dynamics.

For the implementation of the experimental setup, and for the identification of parameters, the flight experiments where conducted with a two-layer controller. A PID controller was designed for the attitude, and a PID controller with gravity compensation was designed for the position. So, the output $F_0$ defined by the position controller is given by [36]

$$F_0 = [0 \quad 0 \quad -m_t g]^T + K_{0p}(p_{0r} - p_0) - K_{0d}\dot{p}_0 + K_{0i}\int(p_{0r} - p_0)dt\,, \tag{56}$$

for which $m_t$ is the total mass of the system, acting on the center of mass of the multicopter. The control gains are positive definite diagonal matrices. The integral term is helpful especially for the vertical motion, to correct steady-state error duo to near-ground effect and degradation of battery power supply.

Similarly, the attitude controller is defined by [36]

$$\tau_\Phi = K_{\Phi p}(\Phi_r - \Phi) - K_{\Phi d}\omega_0 + K_{\Phi i}\int(\Phi_r - \Phi)dt\,, \tag{57}$$

where, again, the control gains are positive definite diagonal matrices. Since in this research the multicopter performs small pitch and roll angles, this controller is based on the approximation $\dot{\Phi} \cong \omega_0$, which is directly measured by the gyroscope sensor. The integral term, in this case, is very important, because multicopters are not perfectly balanced. For example, the hexacopter of this research is modeled as a regular hexagon, with propellers applying thrust on the hexagon perpendicularly to its plane. The center of mass position is modeled to be exactly in the geometrical center, and the inertial sensors to be perfectly aligned with the body-fixed frame. However, none of these assumptions are exactly true, which makes the integral term so important to correct any steady state error. The force in the inertial frame is obtained from the thrust force in the body-fixed frame and the orientation of the multicopter as follows:

$$F_0 = \begin{bmatrix} F_{0x} \\ F_{0y} \\ F_{0z} \end{bmatrix} = R_{i0} \begin{bmatrix} 0 \\ 0 \\ -F_t \end{bmatrix} = -F_t \begin{bmatrix} \sin\phi \sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi) \\ -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi) \\ \cos(\phi)\cos(\theta) \end{bmatrix}. \quad (58)$$

The block *Attitude/Thrust* is defined by rearranging Equation (58) in terms of $\theta_r$, $\phi_r$, and $F_t$, which results in [3]

$$\begin{aligned} \theta_r &= \arctan\left( \frac{F_{0x}\cos(\psi) + \mathrm{F}_{0y}\sin(\psi)}{F_{0z}} \right) \\ \phi_r &= \arctan\left( \frac{F_{0x}\sin(\psi) - F_{0y}\cos(\psi)}{F_{0z}} \cos(\theta_r) \right) \\ \mathrm{F}_t &= -\frac{F_{0z}}{\cos(\phi_r)\cos(\theta_r)}. \end{aligned} \quad (59)$$

The most critical part in an aerial manipulator control, when compared to a typical multicopter, is the attitude control, because the robotic arm dynamics affects the attitude of the multicopter and the translation controller is affected by attitude errors. The dynamics of the robotic arm disturbs the attitude of the multicopter in several different ways. First, the moment of inertial of the vehicle varies for different arm positions, which means that if a constant gain controller is implemented, the system will not have the same performance for all arm positions. Actually, if the multicopter is stable for a given condition, it does not guarantee that it is stable for another arm position. Second, if the arm is sufficiently heavy and moving fast, centrifugal and Coriolis effects disturb the attitude controller. Additionally, a gravitational torque is also originated from the change of the center of mass of the robotic arm. Finally, when torques are applied by the robotic arm servos, torques are induced on the center of mass of the multicopter. So, unless the robotic arm is considerably light and moves slowly, some or all of those effects have to be considered when implementing the control law.

The control law is designed to globally linearize the dynamic system. For that an auxiliary input $\sigma = \begin{bmatrix} \sigma_p & \sigma_\Phi & \sigma_\alpha \end{bmatrix}^T$ is defined such that [3]

$$\tau = \begin{bmatrix} F_0 \\ \tau_\Phi \\ \tau_\alpha \end{bmatrix} = B(q)\begin{bmatrix} \sigma_p \\ \sigma_\Phi \\ \sigma_\alpha \end{bmatrix} + C(q,\dot{q})\dot{q} + G(q) \ . \tag{60}$$

In Equation (60), the attitude is controlled with consideration of the full dynamics of the system by defining the control law in terms of $\sigma_\Phi$ to compute $\tau_\Phi$. Once the vector $\sigma_\Phi$ is known and $F_0$ is also known from the position controller, either $\tau_\alpha$ or $\sigma_\alpha$ must be known to define a linear system of *6+n* equations and *6+n* variables for the determination of $\tau_\Phi$. The servos used in this research are not controlled by torque directly. Instead, the goal position is sent to the robot and the torque is provided by an internal controller. Additionally, the servos do not provide an accurate torque feedback. Then, since the torque $\tau_\alpha$ is not known, $\sigma_\alpha$ must be determined to solve Equation (60). In order to determine $\tau_\Phi$ from $F_0$, $\sigma_\Phi$, and $\sigma_\alpha$, the dynamic equation is written as

$$\begin{bmatrix} I_{3x3} & 0 & 0 \\ 0 & I_{3x3} & 0 \\ 0 & 0 & I_{nxn} \end{bmatrix}\begin{bmatrix} F_0 \\ \tau_\Phi \\ \tau_\alpha \end{bmatrix} = \begin{bmatrix} B_{pp} & B_{p\Phi} & B_{p\alpha} \\ B_{\Phi p} & B_{\Phi\Phi} & B_{\Phi\alpha} \\ B_{\alpha p} & B_{\alpha\Phi} & B_{\alpha\alpha} \end{bmatrix}\begin{bmatrix} \sigma_p \\ \sigma_\Phi \\ \sigma_\alpha \end{bmatrix} + C\dot{q} + G \tag{61}$$

where the term in the left side is multiplied by the identity matrix and the matrix *B* is fragmented in its minor terms for algebraic manipulation as follows:

$$\begin{bmatrix} -B_{pp} & 0 & 0 \\ -B_{\Phi p} & I_{3x3} & 0 \\ -B_{\alpha p} & 0 & I_{nxn} \end{bmatrix}\begin{bmatrix} \sigma_p \\ \tau_\Phi \\ \tau_\alpha \end{bmatrix} = \begin{bmatrix} -I_{3x3} & B_{p\Phi} & B_{p\alpha} \\ 0 & B_{\Phi\Phi} & B_{\Phi\alpha} \\ 0 & B_{\alpha\Phi} & B_{\alpha\alpha} \end{bmatrix}\begin{bmatrix} F_0 \\ \sigma_\Phi \\ \sigma_\alpha \end{bmatrix} + C\dot{q} + G \tag{62}$$

and the torque $\tau_\Phi$ is determined by

$$\begin{bmatrix} \sigma_p \\ \tau_\Phi \\ \tau_\alpha \end{bmatrix} = \begin{bmatrix} -B_{pp} & 0 & 0 \\ -B_{\Phi p} & I_{3x3} & 0 \\ -B_{\alpha p} & 0 & I_{nxn} \end{bmatrix}^{-1}\left(\begin{bmatrix} -I_{3x3} & B_{p\Phi} & B_{p\alpha} \\ 0 & B_{\Phi\Phi} & B_{\Phi\alpha} \\ 0 & B_{\alpha\Phi} & B_{\alpha\alpha} \end{bmatrix}\begin{bmatrix} F_0 \\ \sigma_\Phi \\ \sigma_\alpha \end{bmatrix} + C\dot{q} + G\right). \tag{63}$$

In order to apply Equation (63) to determine $\tau_\Phi$, the value of $\sigma_\alpha$ must be known. One way to determine $\sigma_\alpha$ is to differentiate the angular velocity, since the robotic arm provides position and velocity feedbacks. However, it results in a signal with a high noise level and, for this reason, $\sigma_\alpha$ is obtained in a different way. Since the kinematics of the robotic arm is known, a dynamic observer is designed to estimate the robotic arm accelerations based on the angular goal position for each joint. In Figure 24, a system with equation $\ddot{\alpha} = \sigma$ is designed. The input to the estimator is the same goal position that is sent to the robotic arm. The system is implemented to produce the same kinematic behavior of the robotic arm, which means that this system tracks the state variables of the robotic arm.

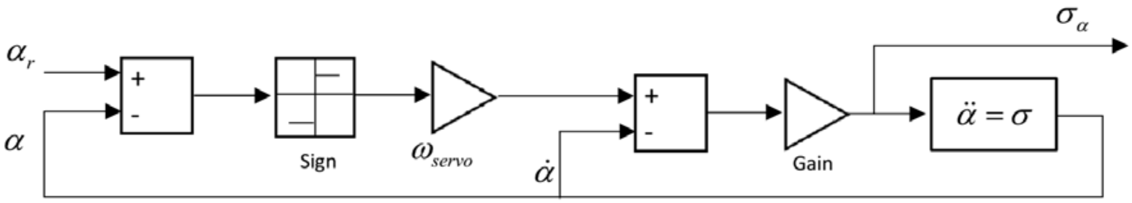


Figure 24. Acceleration Estimator for a Single Joint

In Figure 25, the result of the estimator is depicted for the motion of a robotic servo when the joint rotates from $\pi / 2$ rad to zero. The same motion is experimentally measured and plotted in Figure 22. The estimated angle and angular velocity are consistent with the experimental measurements. Since the differentiation of the angular velocity feedback is very noisy, it is not possible to know the exact accelerations of the joints and it is not possible to design an estimator that reproduces the angular accelerations exactly. On the other hand, the velocity variation when the servo stops or starts moving determines the acceleration impulse. Although the pulse width may be different, the impulse, or the area under the acceleration curve, is the same for the servo and the estimator. With this estimator, the multicopter reacts to the arm movement with the same torque impulse. Another possibility is to neglect the joints angular accelerations and make $\sigma_\alpha = 0$, since during most of the time its value is zero. In this case, when the servo torque impulse is induced on the multicopter, the resultant acceleration is damped by the attitude controller. It works well when the arm is not heavy and the induced torques are relatively small, and reduces the processing load in the microcontroller. Simulations are carried out to verify the impact of neglecting these accelerations for the aerial manipulator in Chapter VI.

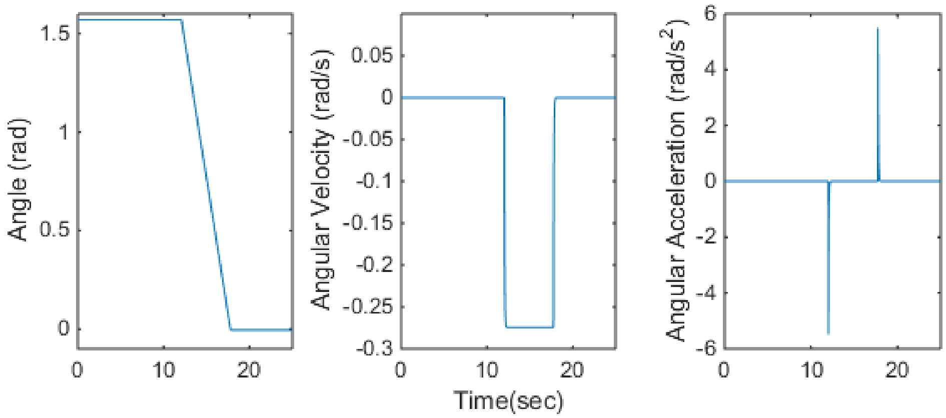


Figure 25. Estimator Output for a Single Joint

In Figure 26, the multicopter controller is changed to incorporate the dynamics of the robotic arm and improve the performance or the free-flight controller. The outer layer is changed to include the goal positions $\alpha_r$ of the robotic arm. The inner layer is presented in more details in Figure 27, where a PD controller outputs the auxiliary variable $\sigma_\Phi$. As from Equation (40), the torque applied by the rotors in the body-fixed frame is defined by $\tau_p = Q^{-T}\tau_\Phi$. The integral term is added to $\tau_p$ directly, since it is designed to compensate errors that affect the torque directly, like near-wall effect, incorrect sensor alignment, and imperfect center of mass balancing.

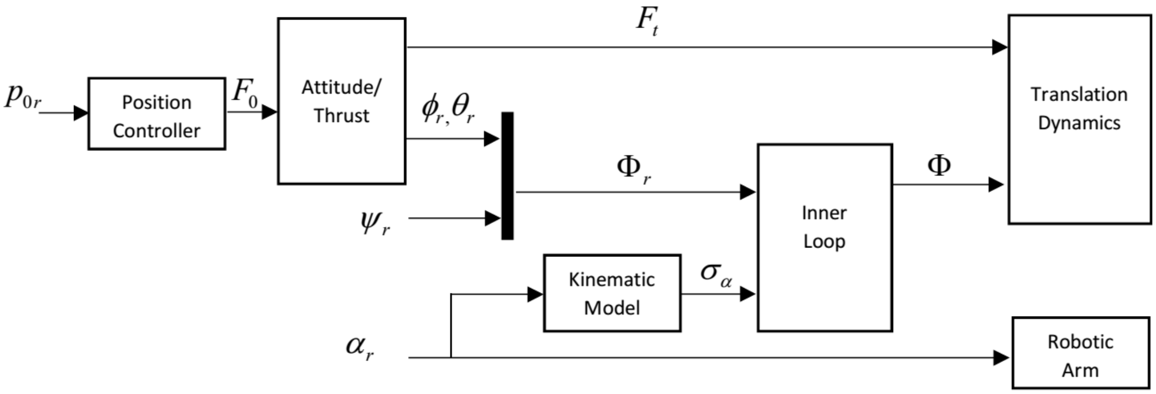


Figure 26. Aerial Manipulator Free-Flight Controller

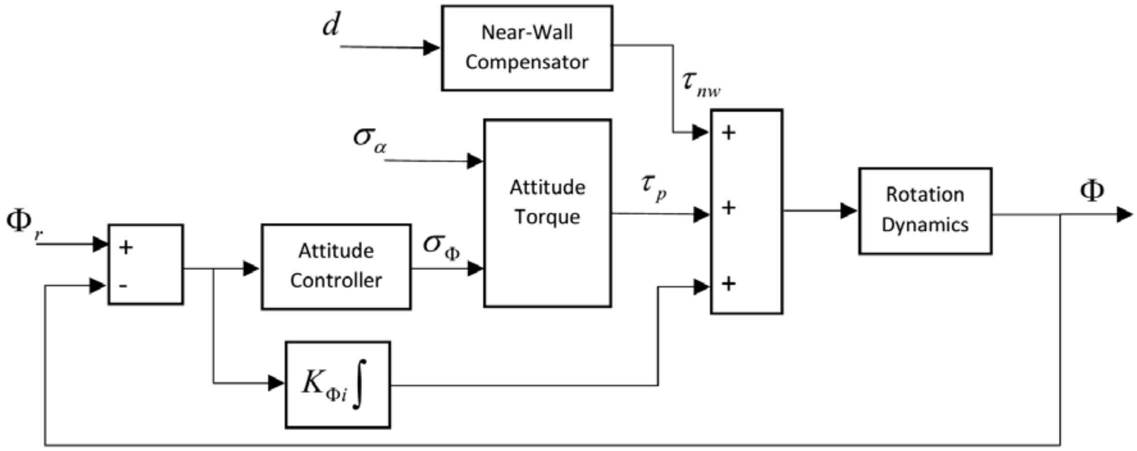


Figure 27. Aerial Manipulator Free-Flight Inner-Loop Controller

The near-wall effect study considers the case of a vehicle approaching a vertical wall perpendicularly. The position of the wall with respect to the multicopter is known. In the laboratory, this relative position is measured by the motion capture system. In outdoor applications it could be measured by means of range sensors. With the experiments presented in Chapter VII, a curve $\tau_{nw}(d)$ is obtained that characterizes the near-wall torque $\tau_{nw}$ as function of the distance $d$ from the wall. In Figure 27, the block that represents the near-wall torque estimator is included in the model.

After the inputs to the hexacopter are obtained, in terms of the total trust and the torques applied on its center of mass, a mixer generates the appropriate PWM signals to be transmitted to the motors. From Equation (45), the PWM signals are related linearly with the thrust force and the torques. So, this equation can be simplified to

$$\begin{bmatrix} F_t \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = A_{PWM} V_{PWM} + B_{PWM} \tag{64}$$

where the vector $V_{PWM} = \left[ PWM_1 \quad PWM_2 \quad PWM_3 \quad PWM_4 \quad PWM_5 \quad PWM_6 \right]^T$, $A_{PWM}$ is a constant matrix, and $B_{PWM}$ is a constant vector. Since there are six PWM signals to control a four-dimensional vector, the mixer is redundant. In order to choose one possible solution, the vector $V_{PWM}$ is obtained in terms of the pseudoinverse of the matrix $A_{PWM}$. By defining

$$\begin{gathered} A_{mix} = A_{PWM}{}^T \left( A_{PWM} A_{PWM}^T \right)^{-1} \\ B_{mix} = -A_{PWM}{}^T \left( A_{PWM} A_{PWM}^T \right)^{-1} B_{PWM} \end{gathered} , \tag{65}$$

the PWM signals are finally determined by

$$V_{PWM} = A_{mix} \begin{bmatrix} F_t \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} + B_{mix} \ . \tag{66}$$

For the propellers used in this study, the PWM signals vary from 1100$\mu$s to 1900$\mu$s. If the signals saturate, the force and torques actually applied by the thrusters are different from those in Equation (66), and unexpected behavior may be observed. So, it is very important to design the controller gains so that the PWM signals do not saturate.

## B. EQUILIBRIUM-BASED FORCE/TORQUE CONTROLLER

In this section, controllers for aerial manipulators under interaction with an object on the wall are proposed. Initially, an open-loop controller is designed based on the equilibrium constraint when the robotic arm interacts with the object. Next, in order to enhance the system stability, the feedback from an internal attitude estimator and a gyroscope sensor, both mounted on the multicopter, is included in the control law. Finally, the system is augmented with a force/torque sensor mounted on the end-effector to design a closed-loop controller. Throughout the development, the equations are derived for a generic aerial manipulator for later application on the experimental vehicle used in this study.

For static interaction between an aerial manipulator and a wall, since multicopters are underactuated and the manipulated object is statically fixed on the wall, two constraints must be satisfied. Since the system is underactuated, the aerial manipulator needs to achieve an appropriate posture to exert a desired force/torque. For example, if the desired interaction is a static force towards the wall, the vehicle is required to have a pitch angle so that a horizontal component of the thrust force is induced, which is visualized in Figure 28. Additionally, in this case, the end-effector manipulates an object that is fixed on the wall. The orientation and position of the end-effector do not vary during the interaction.

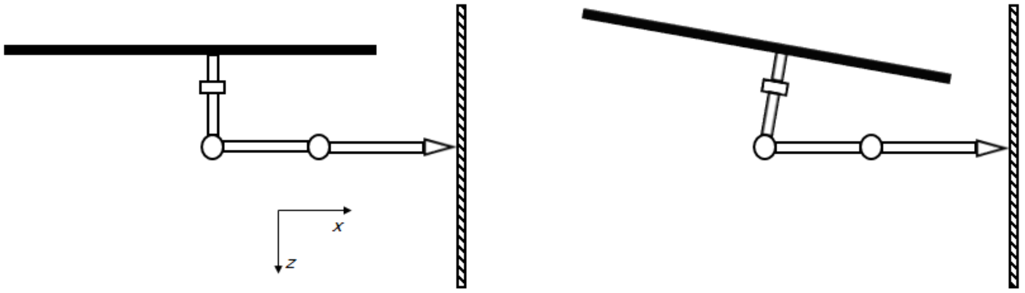


Figure 28. Vertical Surface Interaction

The appropriate posture to achieve the desired force/torque interaction corresponds to an equilibrium condition, where the velocities and accelerations are null. Rewriting the dynamic equation of the system with $\dot{q}=0$ and $\ddot{q}=0$ results in the equilibrium equation for the aerial interaction as follows:

$$G(q)=\tau+\tau_{ext}\ . \tag{67}$$

Substituting Equation (39) and Equation (40) into Equation (67) gives

$$\begin{bmatrix} \begin{bmatrix} 0 & 0 & -F_t \end{bmatrix}^T \\ \tau_p \\ \tau_a \end{bmatrix} = \begin{bmatrix} R_{i0}(q) & 0 & 0 \\ 0 & Q^T(q) & 0 \\ 0 & 0 & I_{n\times n} \end{bmatrix}^{-1} \left( G(q) - J_e^T(q) \begin{bmatrix} F_e \\ \tau_e \end{bmatrix} \right). \tag{68}$$

For the desired interaction, defined by $F_e$ and $\tau_{e,}$ the equilibrium condition is only achieved if the first two elements of the resultant column vector are zero, as in the left-hand side of Equation (68). The first two rows are zero because the thrust force has no components in the *xy* plane of the body-fixed frame. The derivation of the attitude constraint is developed by algebraic manipulation of the first three rows of the equilibrium equation. For that, the relevant components of $G(q)$ and $J_e^T(q)$ are

$$G(q) = \begin{bmatrix} 0 \\ 0 \\ -m_t g \\ \vdots \end{bmatrix} \tag{69}$$

and

$$J_e^T(q) = \begin{bmatrix} I_{3x3} & 0_{3x3} \\ \vdots & \vdots \end{bmatrix}. \tag{70}$$

So, the first three rows of Equation (68) are equivalent to

$$-\begin{bmatrix} \cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi) \\ \cos(\phi)\sin(\theta)\sin(\psi) - \sin(\phi)\cos(\psi) \\ \cos(\phi)\cos(\theta) \end{bmatrix} F_t = \begin{bmatrix} 0 \\ 0 \\ -m_t g \end{bmatrix} - \begin{bmatrix} F_{ex} \\ F_{ey} \\ F_{ez} \end{bmatrix}. \tag{71}$$

If the last row is taken to write $F_t$ in terms of $F_{ez}$, the first two rows can be manipulated to result in

$$\begin{aligned} \tan(\theta) &= \frac{F_{ex}\cos(\psi) + \mathrm{F}_{ey}\sin(\psi)}{F_{ez} + m_t g} \\ \tan(\phi) &= \frac{F_{ex}\sin(\psi) - F_{ey}\cos(\psi)}{F_{ez} + m_t g}\cos(\theta) \end{aligned} \tag{72}$$

which are the final equations for the attitude constraint. It is interesting to note that if $\phi$ and $\theta$ are zero, $F_{ex}$ and $F_{ey}$ are necessarily zero. This is the mathematical confirmation that the multicopter needs to tilt to an appropriate attitude to apply forces in the horizontal plane.

In addition to the underactuation constraint, there is a second constraint related to the orientation of the end-effector during the force/torque interaction. The end-effector orientation $\Phi_{ei}$ is determined from the multicopter orientation $\Phi$ and the joints angles $\alpha$ as follows:

$$\Phi_{ei} = \Phi_e(\eta) \tag{73}$$

where $\eta = [\Phi^T \quad \alpha^T]^T$. Although $\eta$ is variable, $\Phi_{ei}$ is known and constant.

The equilibrium constraints can be grouped into a nonlinear function $f(\eta) = 0$ as follows:

$$f(\eta) = \begin{bmatrix} \Phi_e(\eta) - \Phi_{ei} \\ \tan(\theta) - \dfrac{F_{ex}\cos(\psi) + \mathrm{F}_{ey}\sin(\psi)}{F_{ez} + m_t g} \\ \tan(\phi) - \dfrac{F_{ex}\sin(\psi) - F_{ey}\cos(\psi)}{F_{ez} + m_t g}\cos(\theta) \end{bmatrix} = \begin{bmatrix} \Phi_e(\eta) - \Phi_{ei} \\ f_4(\eta) \\ f_5(\eta) \end{bmatrix} = 0\,. \tag{74}$$

Depending on the complexity of the robotic arm, Equation (74) may not have an analytical solution. For this case, a first order approximation is proposed to compute the solution $\eta_r$ as follows:

$$f(\eta_r) = f(\eta) + F_\eta\left(\eta_r - \eta\right) = 0\,, \tag{75}$$

where

$$F_\eta = \frac{\partial f(\eta)}{\partial \eta}\ . \tag{76}$$

Since $\Phi_{ei}$ is actually a three-by-one vector, the system defined by $f(\eta)$ has five equations and *3+n* variables. If an aerial manipulator has two links with $n = 2$ , $F_\eta$ is a

5x5 square matrix. Equation (75) has a unique solution given by $\eta_r = \eta - F_\eta^{-1} f(\eta)$. If an aerial manipulator has more than two links, the system is redundant. Redundancy can be utilized to accomplish secondary tasks, like increasing reachability, avoiding an obstacle, avoiding singularities, or a achieving a safer posture [2]. In this research, the focus is on accomplishing the primary task of exerting a desired force/torque on a wall. A standard solution using the pseudo-inverse of $F_\eta$ is chosen

$$\eta_r = \eta - F_\eta^T \left( F_\eta F_\eta^T \right)^{-1} f(\eta). \tag{77}$$

Since Equation (77) is a first-order approximation, the solution is valid only if $\eta_r$ is in the neighborhood of the state variables in $\eta$. It is important to analyze and simulate each particular case carefully before the experiments, because the numerical convergence is not guaranteed when $\eta$ is out of the neighborhood of the solution. A better approach is, when planning the experiment, to select a vector $\eta_0$ that is known a priori to be close to the solution and compute Equation (77) for $\eta = \eta_0$.

Let $F_\eta$ be computed from Equation (74) as follows:

$$F_\eta = \begin{bmatrix} \dfrac{\partial \Phi_e}{\partial \eta} \\ \dfrac{\partial f_4(\eta)}{\partial \eta} \\ \dfrac{\partial f_5(\eta)}{\partial \eta} \end{bmatrix}. \tag{78}$$

The partial derivative $\partial \Phi_e / \partial \eta$ can be determined from the Jacobian $J_{\omega e}$, which is known from Equation (12). It is important to note that the end-effector orientation does not depend on the position of the multicopter and the Jacobian can be reduced to a smaller matrix $J_{oe}$ as follows:

$$\omega_e = J_{\omega e} \dot{q} = \left[ 0_{3x3} \quad J_{oe} \right] \left[ \dot{p}^T \quad \dot{\eta}^T \right]^T = J_{oe} \dot{\eta}. \tag{79}$$

Similarly to the multicopter orientation, as in Equation (3), the end-effector angular rates can also be mapped to the time derivatives of the Euler angles

$$\dot{\Phi}_e = Q_e^{-1}\omega_e = Q_e^{-1}J_{oe}\dot{\eta} \tag{80}$$

where $Q_e$ is obtained similarly to the way $Q$ is determined from Equation (4).

Since $\dot{\Phi}_e$ can also be written as

$$\dot{\Phi}_e = \frac{\partial \Phi_e}{\partial \eta}\dot{\eta} \tag{81}$$

the partial derivative of the end-effector orientation is given by

$$\frac{\partial \Phi_e}{\partial \eta} = Q_e^{-1}J_{oe}\,. \tag{82}$$

The partial derivatives of $f_4(\eta)$ and $f_5(\eta)$ are

$$\begin{bmatrix} \dfrac{\partial f_4(\eta)}{\partial \eta} \\ \dfrac{\partial f_5(\eta)}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_4(\Phi)}{\partial \Phi} & \dfrac{\partial f_4(\alpha)}{\partial \alpha} \\ \dfrac{\partial f_5(\Phi)}{\partial \Phi} & \dfrac{\partial f_5(\alpha)}{\partial \alpha} \end{bmatrix} \tag{83}$$

where the derivatives with respect to $\alpha$ are zero, and

$$\begin{bmatrix} \dfrac{\partial f_4(\Phi)}{\partial \Phi} \\ \dfrac{\partial f_5(\Phi)}{\partial \Phi} \end{bmatrix} = \begin{bmatrix} 0 & \dfrac{1}{\cos\left(\theta\right)^2} & \dfrac{F_{ex}\sin(\psi) - \mathrm{F}_{ey}\cos(\psi)}{F_{ez} + m_t g} \\ \dfrac{1}{\cos\left(\phi\right)^2} & \dfrac{F_{ex}\sin(\psi) - F_{ey}\cos(\psi)}{F_{ez} + m_t g}\sin(\theta) & -\dfrac{F_{ex}\cos(\psi) + F_{ey}\sin(\psi)}{F_{ez} + m_t g}\cos(\theta) \end{bmatrix}. \tag{84}$$

An important step during the experiment design is to analyze for what subset of the possible values of $\eta$ the matrix $F_\eta$ is full rank or, in other words, where $F_\eta F_\eta^T$ is invertible. One of the reasons why $F_\eta$ would not be full rank is the impossibility to rotate the multicopter about all possible directions, due to limitations of the robotic arm. For example, if the robotic arm has only one joint, the multicopter only rotates about one axis during manipulation. In this case, the matrix $F_\eta$ has only four columns and the rank is not greater than four. For a robotic arm with at least two degrees of freedom, where the joints are mounted orthogonally to each other, $F_\eta$ is expected to be full rank in most configurations.

To illustrate how to investigate singularities, the specific configuration considered for the simulations and experiments in this research is analyzed. During the experiments, the initial posture from which the multicopter is supposed to apply the forces and torques is shown in Figure 28. For that, the end-effector orientation and robotic arm initial angles are given by

$$\begin{aligned} \Phi_e &= [0 \quad 0 \quad 0]^T \\ \alpha &= [0 \quad \pi/2 \quad 0]^T . \end{aligned} \tag{85}$$

In this case, the desired interaction forces are small when compared to the weight of the aerial manipulator. The attitude angles performed by the multicopter are also small. Equation (77) is calculated from the initial condition $\eta_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & \pi/2 & 0 \end{bmatrix}^T$, which is in the neighborhood of $\eta_r$ where the first order approximation is valid. The partial derivative of $f(\eta)$ is given by

$$F_\eta(\eta_0) = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & \dfrac{-\mathrm{F}_{ey}}{F_{ez} + m_t g} & 0 & 0 & 0 \\ 1 & 0 & \dfrac{-F_{ex}}{F_{ez} + m_t g} & 0 & 0 & 0 \end{bmatrix} \tag{86}$$

which implies

$$\det\left(F_\eta F_\eta^T\right) = 2 \neq 0 . \tag{87}$$

Since the determinant is not zero, the matrix $F_\eta F_\eta^T$ is invertible independently of the desired interaction forces.

In summary, the equilibrium constrains are computed from Equations (68) and (77). First, the reference attitude and joints angles of reference are determined from Equation (77). Once the posture of reference $\eta_r$ is defined, it is substituted in Equation (68) to obtain the thrust and the attitude torques. In Figure 29, the open-loop scheme is shown, where the block *Equilibrium Constraint* represents the calculation of the system inputs based on the desired force/torque interaction. This controller does not use any sensor feedback and, in practice, is not appropriate in terms of stability and convergence. So, another controller is developed to integrate sensor feedbacks to this scheme.

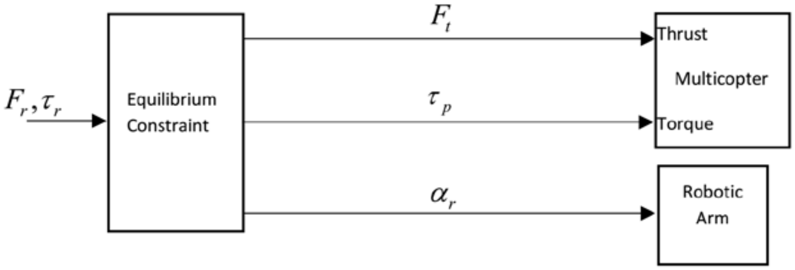


Figure 29. Vertical Surface Interaction Open-Loop Scheme

Although the robotic arm used in this study is modeled as a rigid body, it is built from plastic parts that bend when under force and there is backlash in servo motor gears. When the open-loop controller depicted in Figure 29 was implemented in experiment, rapid oscillations in attitude were observed. For this reason, an attitude PD controller is designed to damp those oscillations as in Figure 30. This controller works similarly to a free-flight attitude controller, in which the attitude is regulated by the rotors to guarantee convergence with stability. The PD attitude control with output to the multicopter is given by

$$\tau_{AC} = K_{\Phi p}(\Phi_r - \Phi) - K_{\Phi d}\omega_0 \tag{88}$$

where the gains are positive definite diagonal matrices.

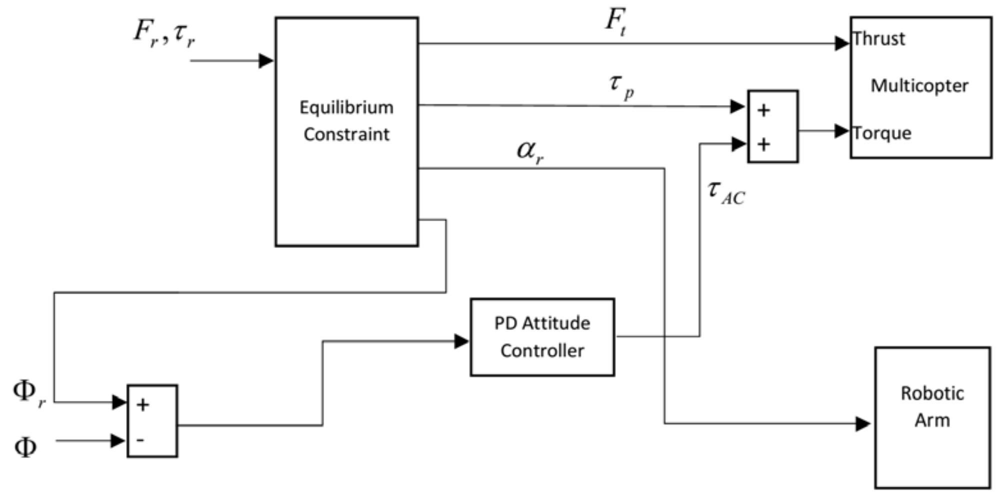


Figure 30. Interaction Closed-Loop Scheme with PD Attitude Controller

The flexibility of the robotic arm is also related to another issue. The robotic arm joints angles are calculated to bring the multicopter to a desired attitude. When the arm bends, it drives the multicopter to an incorrect attitude. When it happens, the PD controller outputs a residual torque because the attitude error is not null. The incorrect attitude and the residual torque cause interaction force and torque errors. To fix this problem, an integral attitude controller is designed, as in Figure 31. With attitude feedback, the robotic arm is moved to a position for which the attitude error is zero. With these two attitude controllers, the multicopter and the robotic arm work in collaboration. The robotic arm angles are modified by an integral controller given by

$$\alpha_{AC} = K_{\Phi i} \int \Delta\alpha dt \tag{89}$$

where the gain $K_{\Phi i}$ is a positive definite diagonal matrix and $\Delta\alpha$ is the attitude error mapped to joints angle errors.

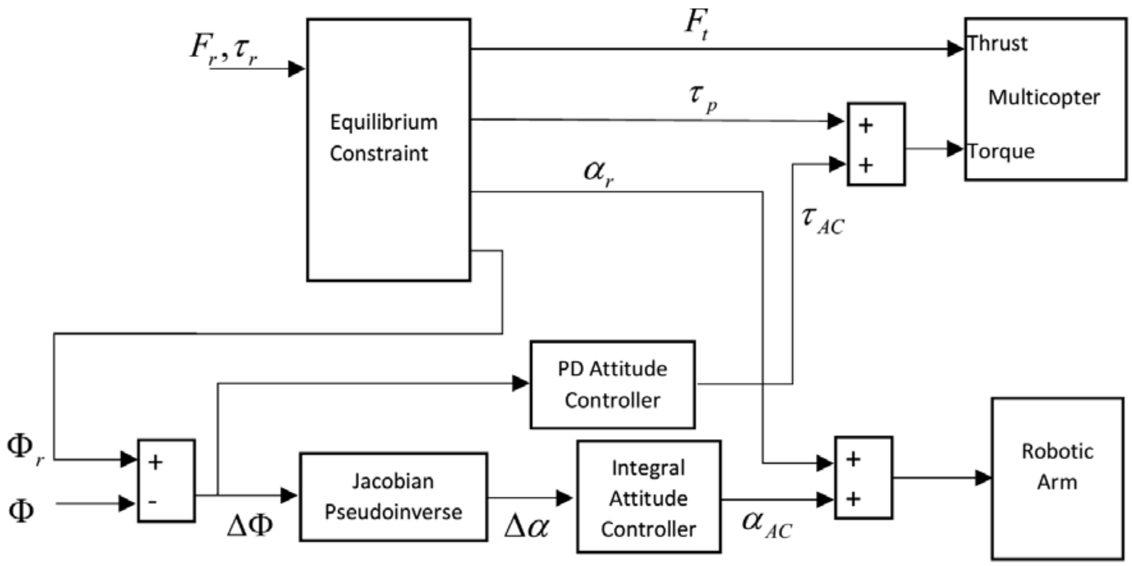


Figure 31. Interaction Closed-Loop Scheme with Integral Attitude Controller

The block *Jacobian Pseudoinverse*, in Figure 31, maps the attitude error to joints angle errors, so that the robotic arm can be controlled to cancel the attitude error. For that, the first step is to find the differential kinematics relation between the robotic arm and the multicopter attitude by rewriting Equation (80) as follows:

$$\dot{\Phi}_e = Q_e^{-1} J_{oe} \begin{bmatrix} \dot{\Phi} \\ \dot{\alpha} \end{bmatrix}. \tag{90}$$

Since the end-effector does not move during interaction, the attitude and the robotic arm joints angles are related by

$$0 = J_{oe} \begin{bmatrix} \dot{\Phi} \\ \dot{\alpha} \end{bmatrix} . \tag{91}$$

For any aerial manipulator, Equation (91) can be rearranged to find a matrix equation that calculates $\dot{\alpha}$ in terms of $\dot{\Phi}$. For redundant robotic arms, the pseudoinverse is used to choose one solution. This is illustrated by the aerial manipulator used in this study. Substituting its matrix $J_{oe}$ into Equation (91) gives

$$0=\begin{bmatrix}1&0&0&1&0&0\\0&1&0&0&1&1\\0&0&1&0&0&0\end{bmatrix}\begin{bmatrix}\dot{\Phi}\\\dot{\alpha}\end{bmatrix}\ \Rightarrow\ \dot{\Phi}=\begin{bmatrix}\dot{\phi}\\\dot{\theta}\\\dot{\psi}\end{bmatrix}=-\begin{bmatrix}1&0&0\\0&1&1\\0&0&0\end{bmatrix}\dot{\alpha}\,. \qquad (92)$$

This equation describes how the attitude of the multicopter is affected by the robotic arm. In this case, the angle $\psi$ does not change while the robot moves. Since the last row is unnecessary, it is removed to eliminate the singularity and then compute the pseudoinverse as follows:

$$\begin{bmatrix}\dot{\phi}\\\dot{\theta}\end{bmatrix}=-\begin{bmatrix}1&0&0\\0&1&1\end{bmatrix}\dot{\alpha}=J_{\Phi\alpha}\dot{\alpha} \qquad (93)$$

$$\dot{\alpha}=J_{\Phi\alpha}^{T}\left(J_{\Phi\alpha}J_{\Phi\alpha}^{T}\right)^{-1}\begin{bmatrix}\dot{\phi}\\\dot{\theta}\end{bmatrix} \qquad (94)$$

$$\dot{\alpha}=-\begin{bmatrix}1&0\\0&0.5\\0&0.5\end{bmatrix}\begin{bmatrix}\dot{\phi}\\\dot{\theta}\end{bmatrix}\,. \qquad (95)$$

Finally, the column correspondent to $\psi$ is included to result in

$$\dot{\alpha}=-\begin{bmatrix}1&0&0\\0&0.5&0\\0&0.5&0\end{bmatrix}\dot{\Phi}\quad\Rightarrow\quad\Delta\alpha\cong-\begin{bmatrix}1&0&0\\0&0.5&0\\0&0.5&0\end{bmatrix}\Delta\Phi\,. \qquad (96)$$

In Figure 32, a force/torque sensor feedback is added to the previous controller to correct modeling errors and external disturbances. It is especially important if high accuracy is required for a manipulation task.

The force controller is implemented through a proportional integral control given by

$$\begin{bmatrix}F\\\tau\end{bmatrix}=K_{p\tau}\left(\begin{bmatrix}F_r\\\tau_r\end{bmatrix}-\begin{bmatrix}F_s\\\tau_s\end{bmatrix}\right)+K_{i\tau}\int\left(\begin{bmatrix}F_r\\\tau_r\end{bmatrix}-\begin{bmatrix}F_s\\\tau_s\end{bmatrix}\right)dt \qquad (97)$$

where the gains $K_{p\tau}$ and $K_{i\tau}$ are positive definite diagonal matrices.

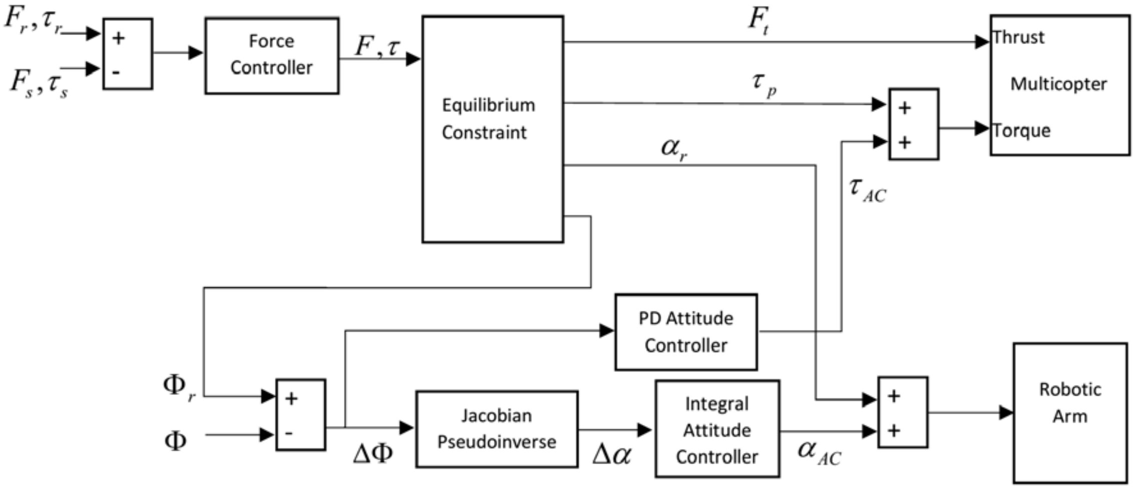


Figure 32. Equilibrium-Based Force/Torque Controller

The equilibrium-based force/torque controller depicted in Figure 32 and the free-flight controller shown in Figure 26 are designed and tested separately. When the aerial manipulator is tasked to interact with objects on a wall, the free-flight controller is initially deployed to approach the wall. As soon as the end effector comes into contact with the wall, the aerial manipulator switches from the free-flight controller to the equilibrium-based force/torque controller. Both controllers are designed based on the dynamic model of the aerial manipulators. For the free-flight stage, the end-effector external forces are zero. For the interaction stage, the inputs are calculated based on an equilibrium point where the accelerations and velocities are zero. Consequently, the same control inputs are generated for an aerial manipulator hovering or interacting with the wall with reference forces equal zero.

# VI. SIMULATION RESULTS

In this chapter the simulation model for the aerial manipulator used in this study and simulation results are presented. Initially, from the dynamic model, the system is written in terms of state-space equations. In Chapter III, the dynamic model of the aerial manipulator is derived from the partial derivatives of the kinetic and potential energies. For the simulator, these partial derivatives are calculated in a recursive manner, with emphasis on computing efficiency. As already mentioned, the parameters of the robotic arm internal controller are not known. For this reason, it is explained how to simulate the arm based on its kinematics. The interactive forces between the robotic arm and a wall are included in the simulation based on a damped spring model. Finally, a virtual environment is developed for a better visualization and understanding of the simulation results. After the full description of the simulation tools, some simulation cases addressing free flight and wall interaction are analyzed.

Let the state variable vector be defined as

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}. \tag{98}$$

Then, the system in the state space is written as follows:

$$\dot{X} = \begin{bmatrix} X_2 \\ \sigma \end{bmatrix} \tag{99}$$

where $\sigma^T = \begin{bmatrix} \sigma_p & \sigma_\Phi & \sigma_\alpha \end{bmatrix}^T$ is determined in terms of the state variables and the inputs, based on the dynamic equation

$$\begin{bmatrix} B_{pp} & B_{p\Phi} & B_{p\alpha} \\ B_{\Phi p} & B_{\Phi\Phi} & B_{\Phi\alpha} \\ B_{\alpha p} & B_{\alpha\Phi} & B_{\alpha\alpha} \end{bmatrix} \begin{bmatrix} \sigma_p \\ \sigma_\Phi \\ \sigma_\alpha \end{bmatrix} + CX_2 + G = \begin{bmatrix} R_{i0} & 0 & 0 \\ 0 & Q^T & 0 \\ 0 & 0 & I_{n\times n} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} 0 & 0 & -F_t \end{bmatrix}^T \\ \tau_p \\ \tau_\alpha \end{bmatrix} + \tau_{ext}\,. \tag{100}$$

As already described in Chapter V, for the robotic arm used in this study, the torque $\tau_\alpha$ is not known, because the servo comes from factory with an unknown internal controller. Since the robot joint accelerations $\sigma_\alpha$ are estimated, the dynamic equation is rearranged to determine $\sigma_p$ and $\sigma_\Phi$ as follows:

$$\begin{bmatrix}\sigma_p\\ \sigma_\Phi\\ \tau_\alpha\end{bmatrix}=\begin{bmatrix}B_{pp} & B_{p\Phi} & 0\\ B_{\Phi p} & B_{\Phi\Phi} & 0\\ B_{\alpha p} & B_{\alpha\Phi} & -I_{n\times n}\end{bmatrix}^{-1}\left(\begin{bmatrix}-R_{i0} & 0 & -B_{p\alpha}\\ 0 & Q^T & -B_{\Phi\alpha}\\ 0 & 0 & -B_{\alpha\alpha}\end{bmatrix}\begin{bmatrix}\begin{bmatrix}0 & 0 & F_t\end{bmatrix}^T\\ \tau_p\\ \sigma_\alpha\end{bmatrix}-CX_2-G+\tau_{ext}\right). \quad (101)$$

The simulation is implemented in Matlab Simulink as in Figure 33. The dynamic model is simulated in the continuous time domain. The state variables are obtained from the integration of $\dot{X}$, which in turn is computed from the state variables in the block *State-Space Model*. The robotic arm estimator is the one presented in Figure 24.

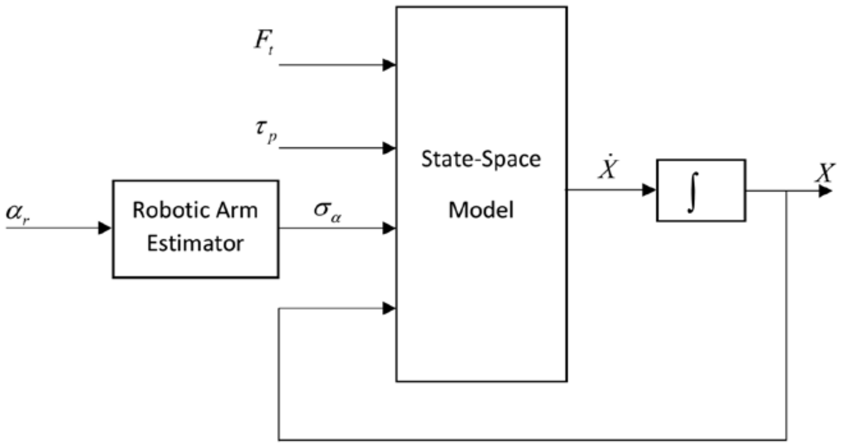


Figure 33. Aerial Manipulator Dynamic Model in Matlab Simulink

The dynamic model derived in Chapter III is obtained in terms of partial derivatives of the potential energy and the kinetic energy. There are several ways to compute the partial derivatives, and the choice depends on the computational tools and the complexity of the system. The aerial manipulator used in this study, for example, is very complex, with nine degrees of freedom interrelated by non-linear expressions. Both the simulations and the controllers are implemented in Matlab Simulink, which is very efficient for matrix numerical computations. In order to have more efficiency, the potential energy and the kinetic energy were computed in Chapter III in a recursive manner. For this reason, the partial derivatives are also computed recursively and maintaining, as much as possible, the matrix notation.

The two terms that depend on partial derivatives are

$$N\left(q,\dot{q}\right)=\begin{bmatrix} \dot{q}^T\,\dfrac{\partial B}{\partial q_1} \\ \vdots \\ \dot{q}^T\,\dfrac{\partial B}{\partial q_{n+6}} \end{bmatrix} \tag{102}$$

$$G\left(q\right)=\begin{bmatrix} \dfrac{\partial P}{\partial q_1} \\ \vdots \\ \dfrac{\partial P}{\partial q_{n+6}} \end{bmatrix}. \tag{103}$$

For each generalized coordinate $q_i$, and for each rigid body $k$, the partial derivatives associated to the potential energy are given by

$$\frac{\partial P_k}{\partial q_i}=\begin{bmatrix}0 & 0 & -m_k.g\end{bmatrix}\frac{\partial p_k}{\partial q_i}. \tag{104}$$

Since the positions of the joints are computed recursively, their partial derivatives are also computed recursively in the same sequence, as follows:

$$\begin{aligned} \frac{\partial p_{s,k+1}}{\partial q_i}&=\frac{\partial p_{sk}}{\partial q_i}+\frac{\partial R_{\imath k}}{\partial q_i}\begin{bmatrix}l_k & 0 & 0\end{bmatrix}^T \\ \frac{\partial p_k}{\partial q_i}&=\frac{\partial p_{sk}}{\partial q_i}+\frac{\partial R_{\imath k}}{\partial q_i}\begin{bmatrix}l_{ck} & 0 & 0\end{bmatrix}^T. \end{aligned} \tag{105}$$

Similarly, for each generalized coordinate $q_i$, and for each rigid body $k$, the partial derivatives associated to the kinetic energy are given by

$$\frac{\partial B_k}{\partial q_i} = \frac{\partial J_k^T}{\partial q_i} M_k J_k + J_k^T M_k \frac{\partial J_k}{\partial q_i} \ . \tag{106}$$

The partial derivatives of the Jacobian $J_k$ are computed as follows:

$$\begin{aligned}
\frac{\partial J_{\omega k}}{\partial q_i} &= \frac{\partial R_{k,k-1}}{\partial q_i} J_{\omega,k-1} + R_{k,k-1} \frac{\partial J_{\omega,k-1}}{\partial q_i} \\
\frac{\partial J_{s,k+1}}{\partial q_i} &= \frac{\partial J_{sk}}{\partial q_i} - \frac{\partial R_{\iota k}}{\partial q_i} \begin{bmatrix} l_k \\ 0 \\ 0 \end{bmatrix}^{\times} J_{\omega k} - R_{\iota k} \begin{bmatrix} l_k \\ 0 \\ 0 \end{bmatrix}^{\times} \frac{\partial J_{\omega k}}{\partial q_i} \\
\frac{\partial J_{pk}}{\partial q_i} &= \frac{\partial J_{sk}}{\partial q_i} - \frac{\partial R_{\iota k}}{\partial q_i} \begin{bmatrix} l_{ck} \\ 0 \\ 0 \end{bmatrix}^{\times} J_{\omega k} - R_{\iota k} \begin{bmatrix} l_{ck} \\ 0 \\ 0 \end{bmatrix}^{\times} \frac{\partial J_{\omega k}}{\partial q_i} \\
\frac{\partial J_k}{\partial q_i} &= \begin{bmatrix} \dfrac{\partial J_{pk}}{\partial q_i} \\ \dfrac{\partial J_{\omega k}}{\partial q_i} \end{bmatrix}.
\end{aligned} \tag{107}$$

The partial derivatives relative to the battery, which is considered a separate rigid body in this research, is given by

$$\begin{aligned}
\frac{\partial P_b}{\partial q_i} &= \begin{bmatrix} 0 & 0 & -m_b.g \end{bmatrix} \frac{\partial p_b}{\partial q_i} \\
\frac{\partial B_b}{\partial q_i} &= \frac{\partial J_b^T}{\partial q_i} M_b J_b + J_b^T M_b \frac{\partial J_b}{\partial q_i}.
\end{aligned} \ . \tag{108}$$

Finally, the partial derivative of the kinetic energy and the potential energy is the summation of the results for each rigid body, as follows:

$$\begin{aligned}
\frac{\partial B}{\partial q_i} &= \frac{\partial B_b}{\partial q_i} + \sum_{k=0}^{n} \frac{\partial B_k}{\partial q_i} \\
\frac{\partial P}{\partial q_i} &= \frac{\partial P_b}{\partial q_i} + \sum_{k=0}^{n} \frac{\partial P_k}{\partial q_i}.
\end{aligned} \tag{109}$$

The wall interaction is modeled as a damped spring system. When the end-effector is not in contact with the wall, the interaction forces and torques are set to be zero. When the end-effector touches the wall and holds the object, the interactions forces and torques are included in the simulation. In the simulation, a threshold is defined relative to the wall position to designate the contact transition as follows:

$$\begin{cases} \tau_{ext} = 0, & x_e < x_w - \Delta x_{\max} \\ \tau_{ext} = J^T \begin{bmatrix} F_e \\ \tau_e \end{bmatrix}, & x_e \geq x_w - \Delta x_{\max} \end{cases} \tag{110}$$

where $x_w$ is the position of the plane that determines the wall, $x_e$ is the end-effector $x$ coordinate, and $\Delta x_{\max}$ is the maximum wall deformation resulted from the interaction forces during manipulation task. The interaction force $F_e$ and torque $\tau_e$ are from Equation (51).

The analysis of simulation results for a system like an aerial manipulator is not trivial. The plots of the state variables, the inputs, and the interaction forces, give detailed information for quantitative comparisons. However, it is very difficult to understand how all those variables relate to each other and extract all relevant information from the simulations. That is why a Simulink virtual environment is developed. The virtual environment consists of a video output where the aerial manipulator flight is recorded from pre-defined points of view. It is also useful for experiments, since it is possible to reproduce them in virtual environment from recorded data.

The virtual environment tool from Simulink reproduces, in video, the movement of each rigid body given its position and orientation. In Figure 34, the drawings of the hexacopter and the three links of the aerial manipulator used in this research are presented.

In Figure 35, the Simulink scheme is shown where the orientation and position of the aerial manipulator moving parts are sent to the virtual environment tool. In Figure 36, some simulation snapshots are presented to illustrate how the virtual environment tool integrates the links and the hexacopter to reproduce the aerial manipulator. Other objects, like the stick on the wall are placed in the scenario and the videos are created from predefined points of view.

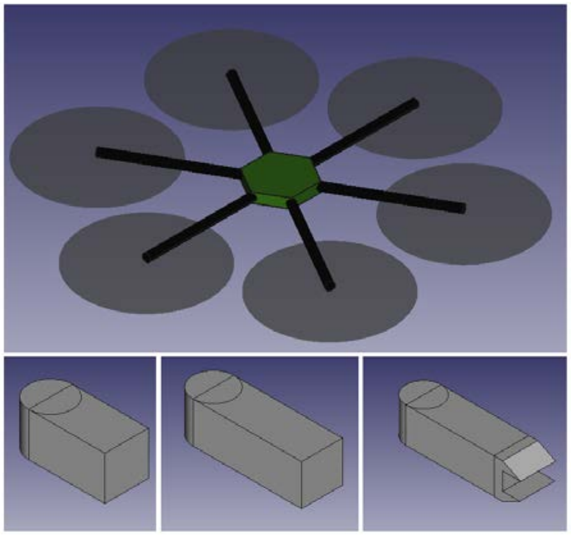


Figure 34. Drawings of Aerial Manipulator Parts

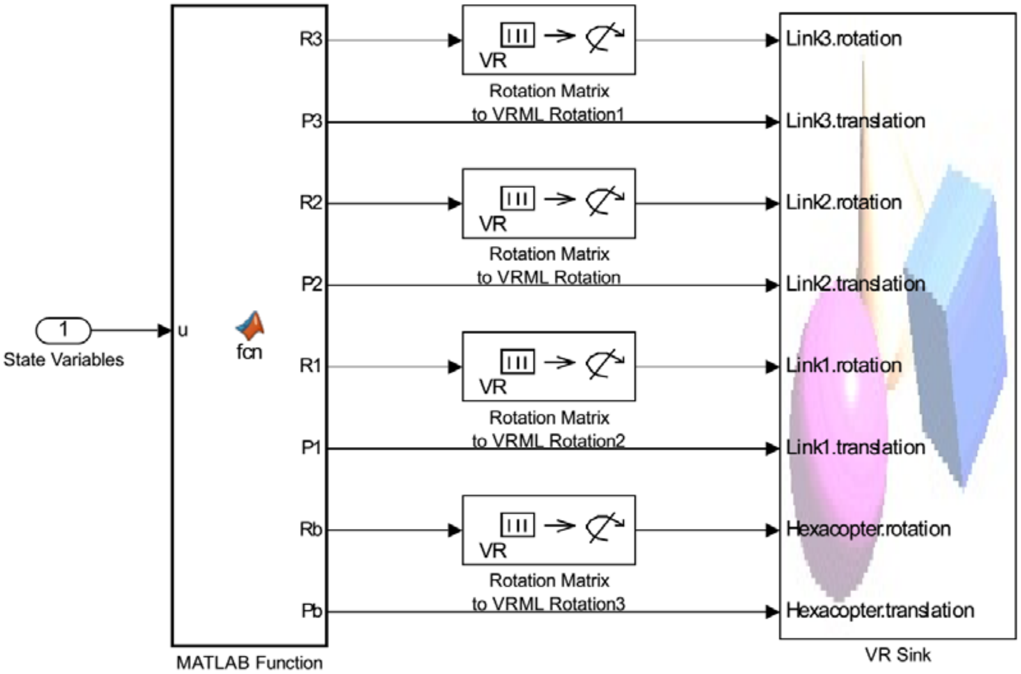


Figure 35. Virtual Environment Implementation in Simulink

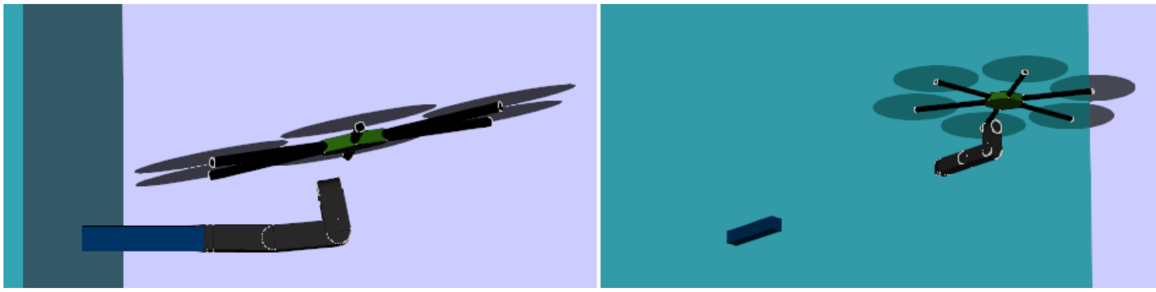


Figure 36. Video Snapshots from Virtual Environment in Simulink

Once the dynamic model and the virtual environment are implemented in Simulink, the simulation file is ready to test the controller. The simulator main blocks are connected as in Figure 37.

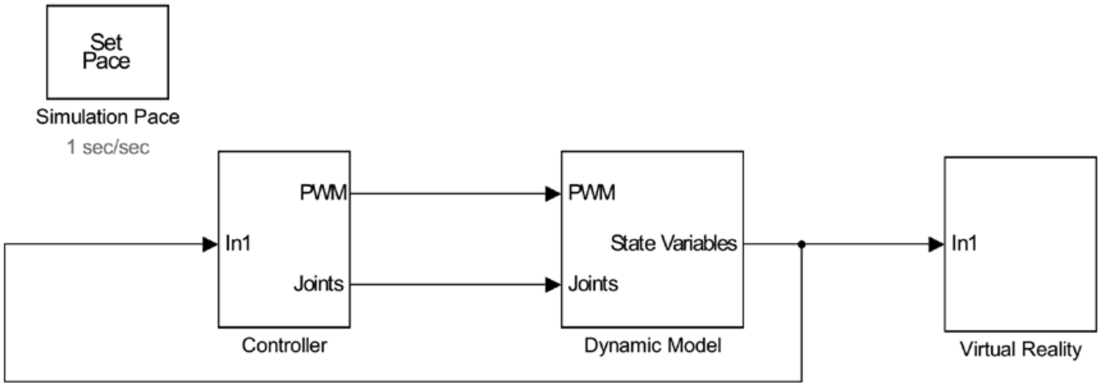


Figure 37. Simulation of the Aerial Manipulator in Simulink

## A. FREE FLIGHT

In this section, simulations are performed for the aerial manipulator free-flight stage. The first simulations are designed to validate the free-flight controller developed in Chapter V. Next, more simulations are designed to investigate the effect of some modifications on flight performance, to adapt the controller to the experiments. The microcontroller used in the experiments is designed for simpler multicopter applications. The aerial manipulator controller is simplified to reduce processing load on the microcontroller. Some terms of the control law are removed to investigate if those terms can be neglected. By removing those terms, a simplified controller is uploaded to the microcontroller. The near-wall effect is addressed experimentally in Chapter VII, but it is not studied in the simulations.

In order to verify the performance of the aerial manipulator flight with a moving robotic arm, a hovering flight with a sequence of robotic arm movements combined with a yaw rotation is performed. The sequence of movements is illustrated in Figure 38, Figure 39, and Figure 40.

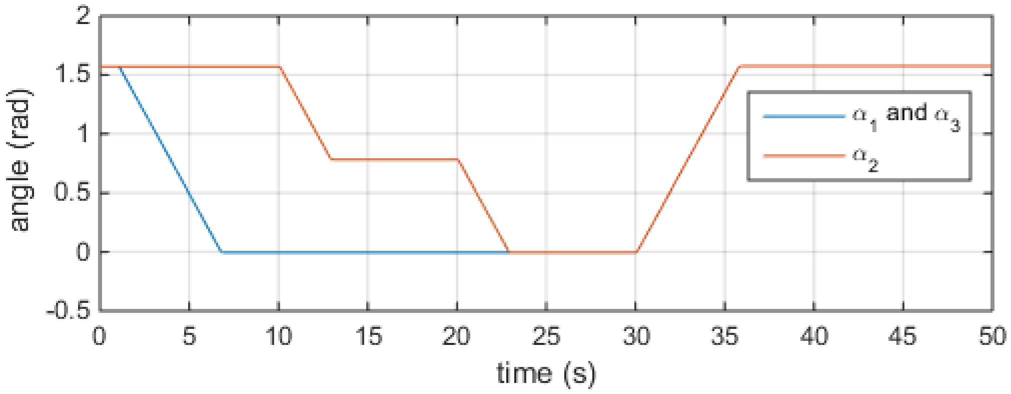


Figure 38. Robotic Arm Angles for the Free-Flight Simulations

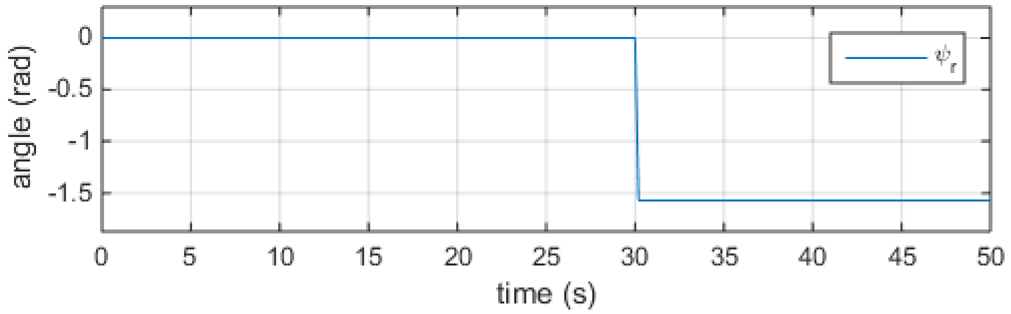


Figure 39. Yaw Reference Angle for the Free-Flight Simulations

The first waypoint replicates the position the aerial manipulator takes off and lands in during the experiments. The retracted arm allows the legs to touch the ground. The second waypoint is achieved by rotating the first and the third joints by 90°. The second joint rotates to 45° to reach the third waypoint. Next, the robotic arm is completely extended in parallel to the vertical axis. Finally, the multicopter performs a yaw rotation while the robotic arm retracts to the initial position. During this sequence of movements, the free-flight simulator is tested while the center of mass and moment of inertia of the aerial manipulator are changing. The last movement combines yaw rotation with arm movement to study the effect of the centrifugal and Coriolis forces.

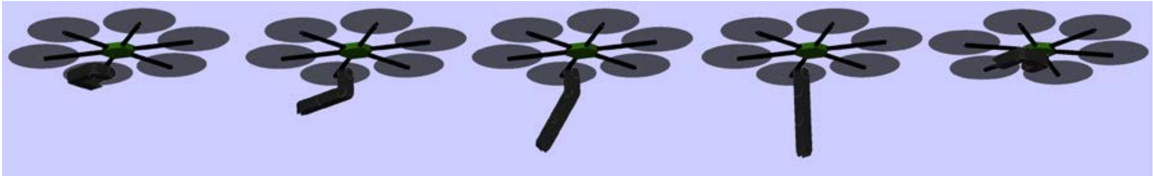


Figure 40. Sequence of Waypoints for the Free-Flight Simulations

### 1. Multicopter Controller without Manipulator Compensation

In order to validate the aerial manipulator free-flight controller, a multicopter controller without manipulator compensation is simulated first to serve as a baseline. This controller is implemented by applying changes on the aerial manipulator control law defined in Equation (60). The matrices $B(q)$, and $G(q)$ are set to be constant and equal to their initial values $B(q_0)$ and $G(q_0)$. The matrix $C$ and the vector $\sigma_\alpha$ are set to be zero. The multicopter controller without compensation and the aerial manipulator controller described in Chapter V are exactly the same, except that the controller without compensation has no knowledge of the robotic arm movements.

The results of the first simulation are presented in Figure 41, Figure 42, and Figure 43. The robotic arm first joint is aligned with the roll axis. In Figure 41, the performance of the roll angle control is worse when the robotic arm first joint is moving. This is an evidence that the multicopter controller performance is better when the robotic arm is not moving. It is evident from the plots of the attitude angles $\theta$ and $\phi$ that, when the robotic arm moves, the multicopter does not follow the reference inputs $\theta_r$ and $\phi_r$ accurately. The center of mass position of the aerial manipulator varies significantly during the robotic arm movement. The attitude controller performance is deteriorated by the variable torque induced on the multicopter body by the weight of the aerial manipulator. Since the attitude controller is not effective, the performance of the position controller is also affected. The position error along the *x* axis reaches almost 80 cm, which is a high number for a multicopter position error.

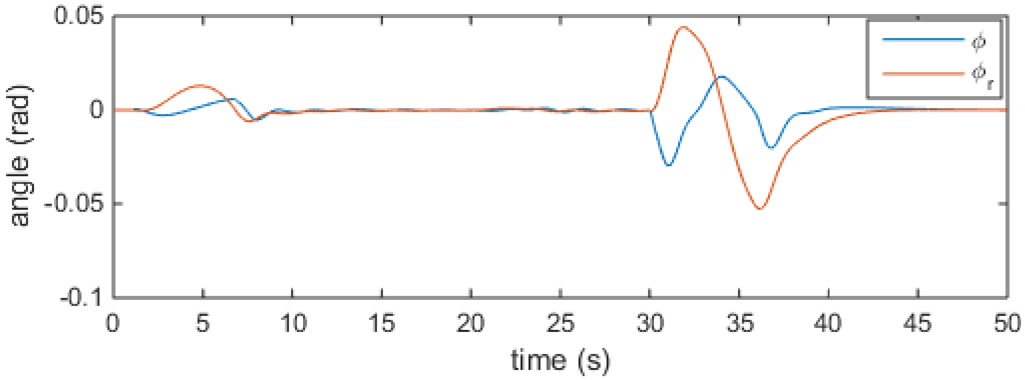


Figure 41. Roll Angle for the Controller without Manipulator Compensation

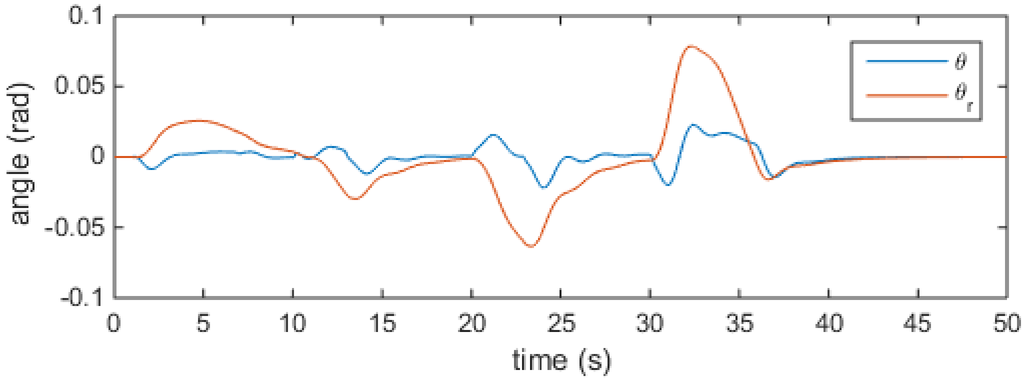


Figure 42. Pitch Angle for the Controller without Manipulator Compensation

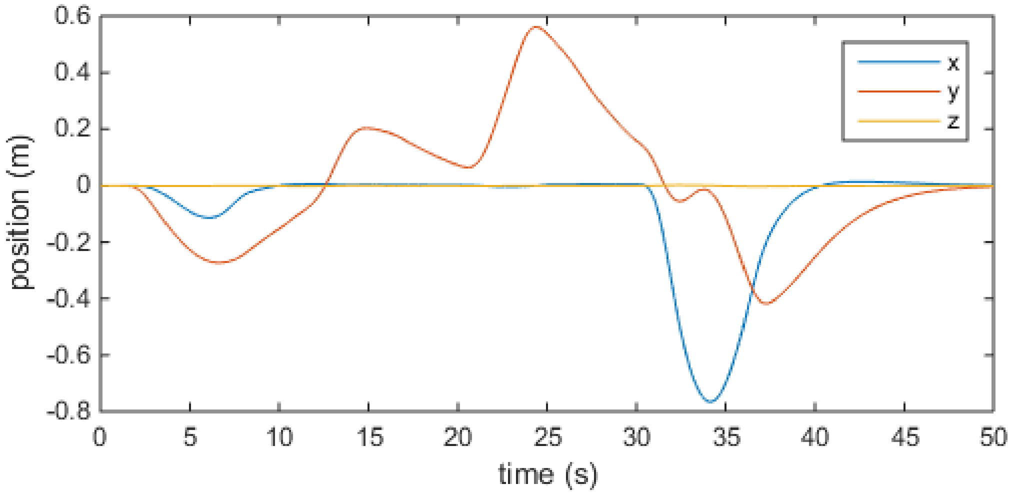


Figure 43. Position for the Controller without Manipulator Compensation

## 2. Aerial Manipulator Free-Flight Controller

The case presented in this section addresses the aerial manipulator free-flight controller described in Chapter V. All the terms of the control law defined in Equation (60) are applied. For this reason, the attitude controller reacts instantaneously to the robotic arm movements. In Figure 44 and Figure 45, the attitude error is practically equal to zero because the multicopter is able track the attitudes of reference efficiently. The position error, in Figure 46, is smaller than position error for the multicopter controller without manipulator compensation by two orders of magnitude.

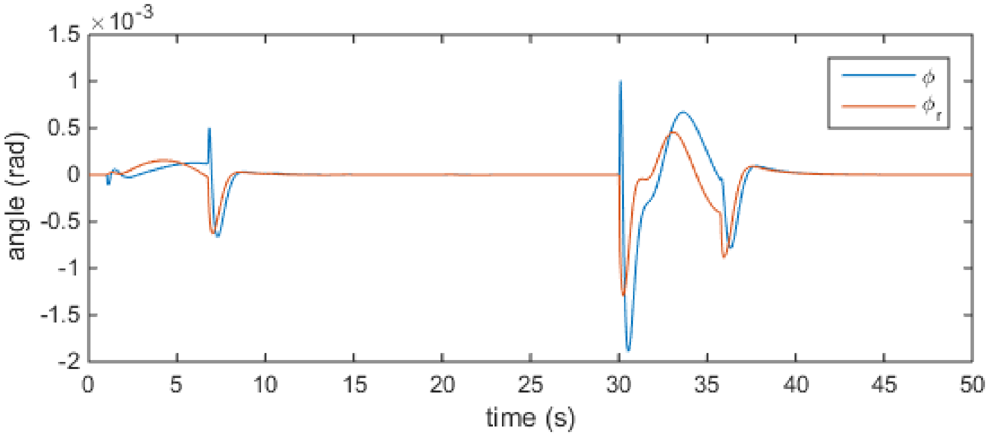


Figure 44. Roll Angle for the Aerial Manipulator Free-Flight Controller

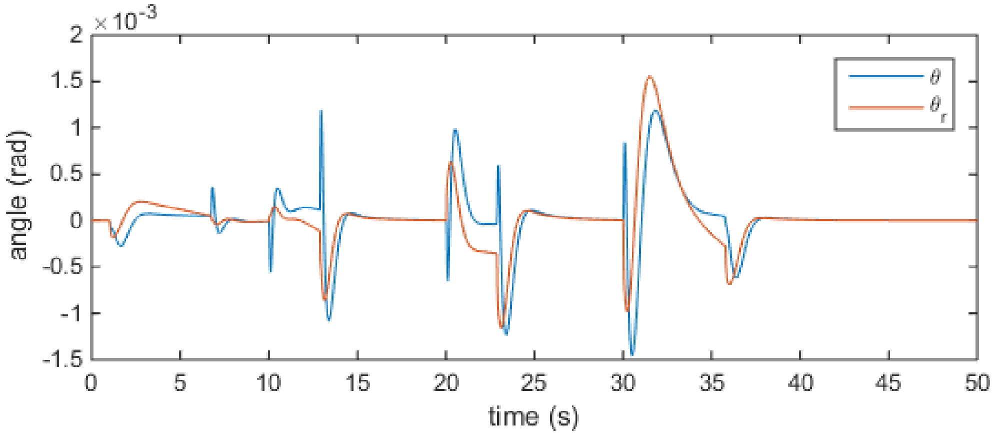


Figure 45. Pitch Angle for the Aerial Manipulator Free-Flight Controller

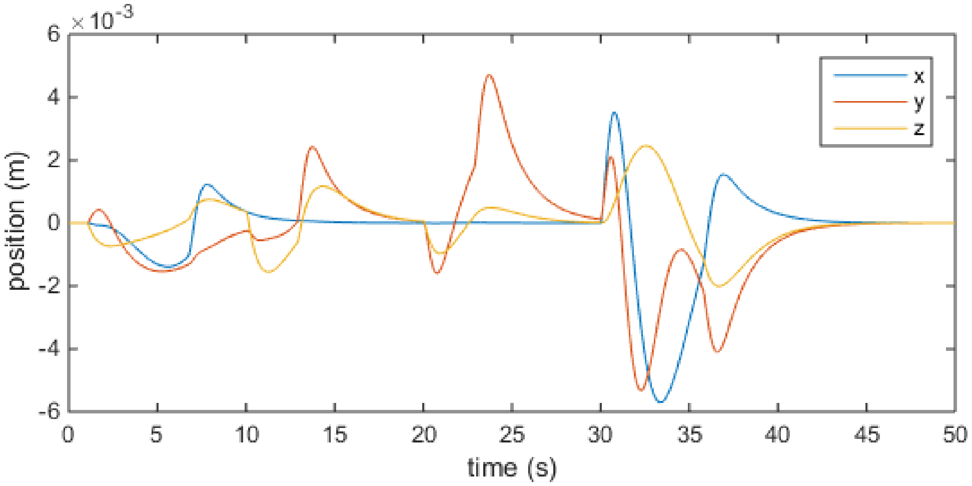


Figure 46. Multicopter Position for the Aerial Manipulator Free-Flight Controller

In Figure 47, the torques applied by the rotors on the multicopter body are shown. Each time the robotic arm starts or stops to move, the joints angular rates change practically instantaneously. The angular acceleration impulses are estimated and included in the control law. Therefore, impulse torques are applied by the rotors to react to the servos acceleration impulses. In Figure 44 and Figure 45, the attitude angles track the reference angles efficiently even when the servos induce torques impulses on the multicopter body. Another aspect of the plot in Figure 47  is that the dominant terms in the attitude control law are the gravity compensation terms. The magnitude of the torques are mainly determined by the posture of the robotic arm. The gravity compensation is extremely important for aerial manipulators flight accuracy.

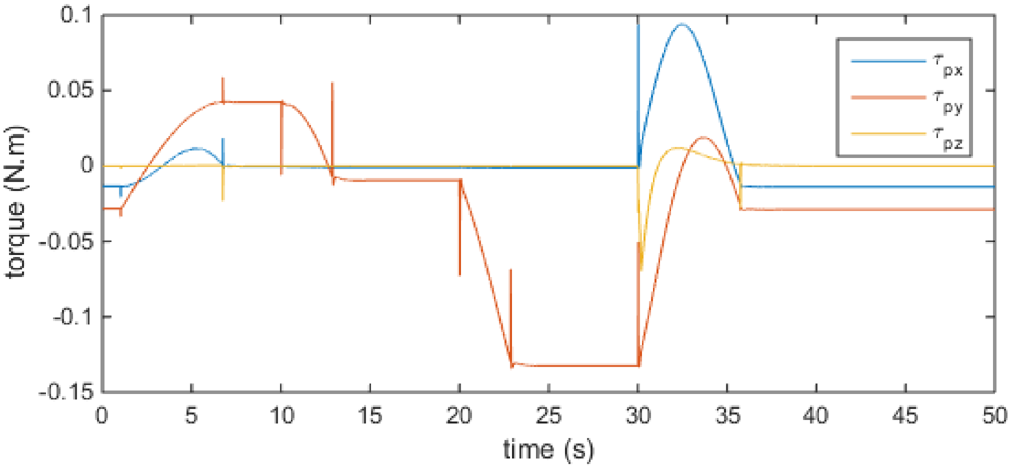


Figure 47. Attitude Torques for the Aerial Manipulator Free-Flight Controller

### 3. Aerial Manipulator Simplified Controller

The aerial manipulator simplified controller is designed to reduce processing load on the microcontroller. The robotic arm is programmed to move slowly during the experiments. For this reason, the centrifugal and Coriolis terms, as well as the estimated angular accelerations of the robotic arm are removed from the control law. Most of the processing load of the complete controller are due to the calculation of the partial derivatives necessary to compute the centrifugal and Coriolis terms. In this section the simplified controller is compared to the complete controller to determine the significance of the neglected terms on the flight performance. In order to implement the aerial manipulator simplified controller, the matrix $C$ and the vector $\sigma_\alpha$ of the control law defined in Equation (60) are set to be zero.

In Figure 48 and in Figure 49, the attitude angles roll and pitch for the simplified aerial manipulator controller are shown. The multicopter position for the aerial manipulator simplified controller is shown in Figure 50. The performance of the simplified controller is slightly worse than the performance of the complete controller.

The sequence of movements is chosen to emphasize the centrifugal and the Corioli forces. For instance, in the transition to the last waypoint, the aerial manipulator rotates about the vertical axis while moving the robotic arm. Although the Corioli and centrifugal forces are neglected in the simplified control law, the position error is still in the order of 1 cm, which is significantly small.

The attitude torques for the simplified aerial manipulator controller are shown in Figure 51. Although the control law neglects the acceleration impulses of the of the joints angles, the attitude torque plot still presents small pulses in reaction to the arm movement. Those small pulses are caused by the PD attitude controller, which reacts immediately to the robotic arm acceleration impulses.

The simplified controller is implemented in the free-flight experiments described in Chapter VII.

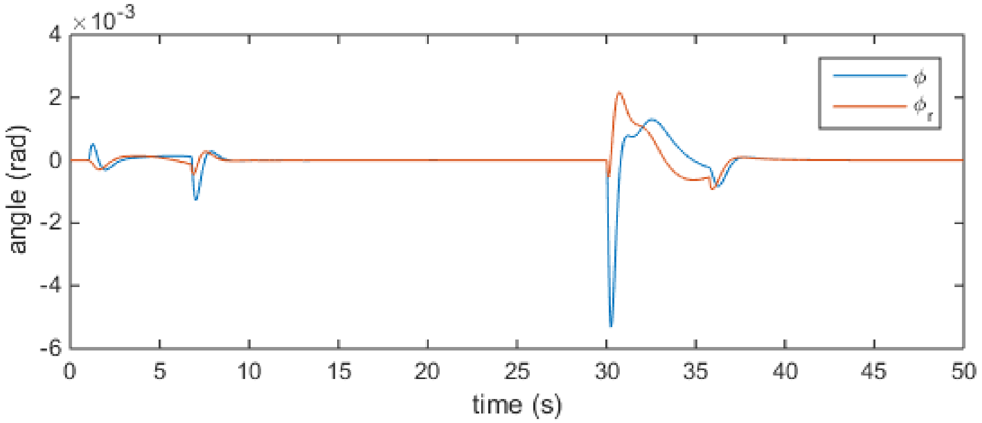


Figure 48. Roll Angle for the Aerial Manipulator Simplified Controller

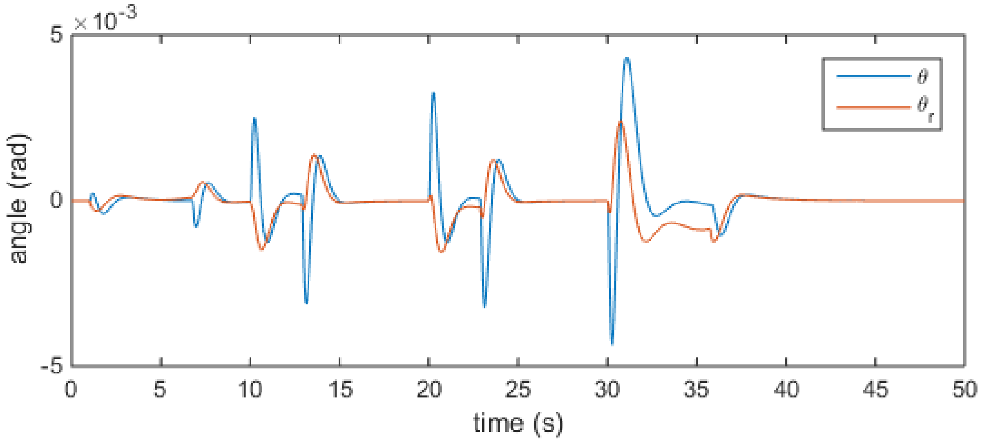


Figure 49. Pitch Angle for the Aerial Manipulator Simplified Controller

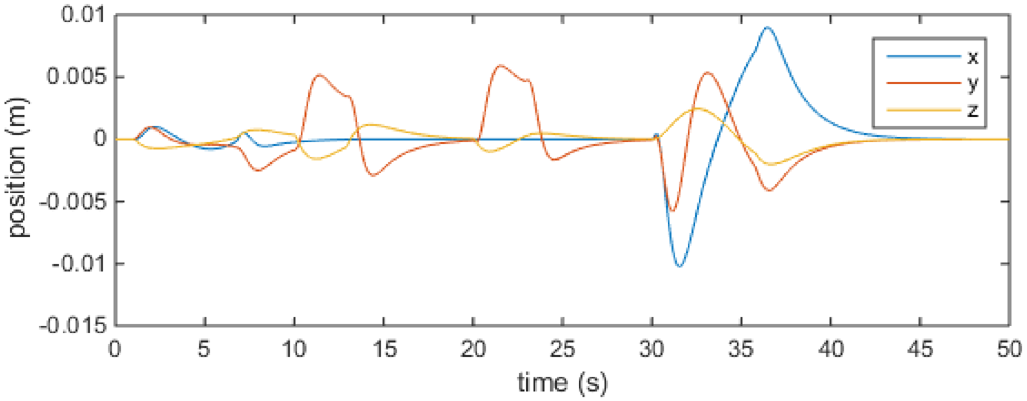


Figure 50. Multicopter Position for the Aerial Manipulator Simplified Controller

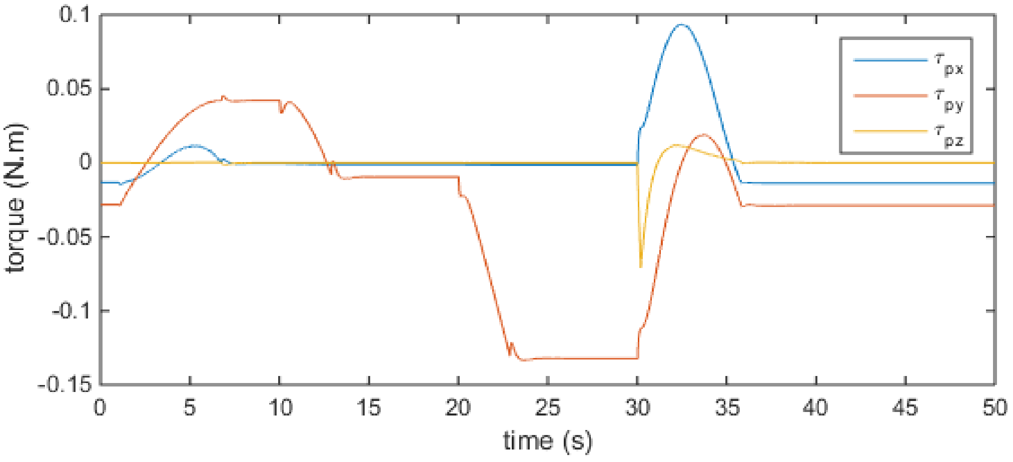


Figure 51. Attitude Torques for the Aerial Manipulator Simplified Controller

## B. EQUILIBRIUM-BASED FORCE/TORQUE CONTROLLER

In this section, the simulations are performed to study the equilibrium-based force/torque controller presented in Chapter V. The experimental setup is reproduced in the simulation environment. The force/torque reference is a six-dimensional vector given by $\tau_r = \begin{bmatrix} F_{rx} & F_{ry} & F_{rz} & \tau_{rx} & \tau_{ry} & \tau_{rz} \end{bmatrix}^T$. In the first simulation cases, a force or a torque is applied in only one direction, to study each component individually. Another simulation case is implemented where components of forces and torques are applied simultaneously. In the last case an interaction without a force-torque sensor feedback is simulated.

The simulation configuration is illustrated in Figure 52. The torques are expressed in terms of the end-effector frame. The external interaction forces are expressed with respect to the inertial frame. The vertical surface belongs to a plane orthogonal to the *y* axis. There is a stick on the wall because the robotic arm is not long enough to reach the wall.

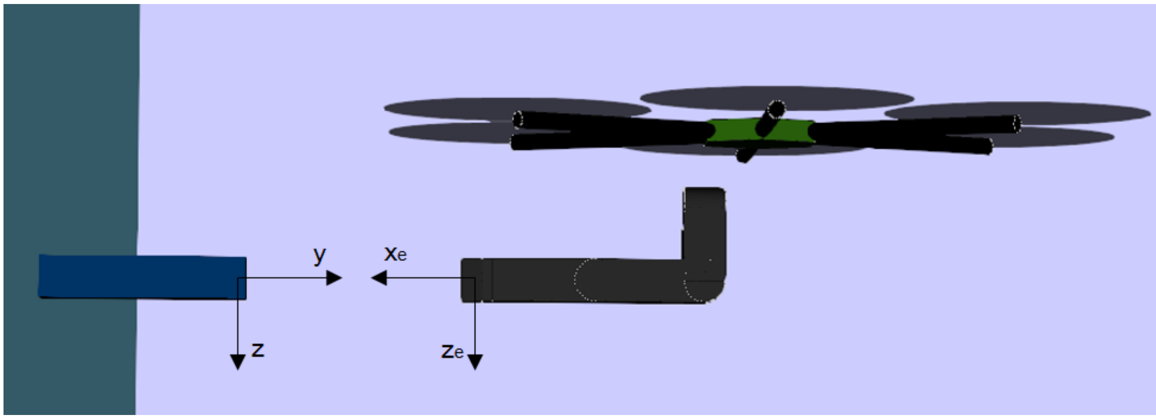


Figure 52. Inertial Frame and End-Effector Frame for the Force/Torque Simulations

### 1. Lateral Force Interaction

The lateral force is implemented by making the reference force/torque $\tau_r = \begin{bmatrix} 1\text{N} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$. The aerial manipulator applies a lateral force on the object on the wall by rolling the moticopter body. The lateral force applied by the thrusters on the multicopter body is transmitted through the robotic arm to the end-effector, which applies a lateral force on the object. In Figure 53, the forces and torques for the lateral force simulation are plotted. The aerial manipulator successfully applies the 1.0 N force along the x axis with stability. However, the other components do not converge to zero.

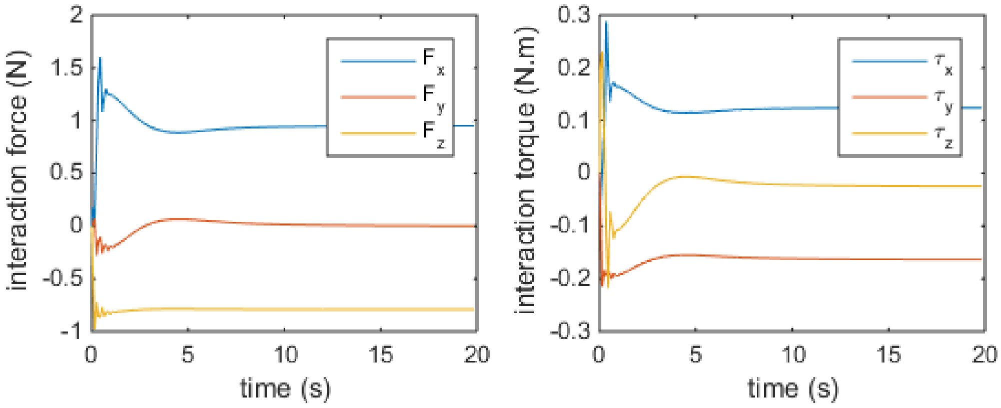


Figure 53. Forces and Torques for the Lateral Force Simulation

When the multicopter rolls to apply the lateral force, the aerial manipulator behaves as a wrench. A vertical interaction torque is consequently applied on the object. In order to cancel this torque, the aerial manipulator force-torque controller applies a vertical torque to the opposite direction. The multicopter torque is limited by the maximum torque applied by each thruster. The vertical torque the multicopter is commanded to apply in this simulation exceeds the maximum limit and the PWM signals saturate, as shown in Figure 54. A PWM signal varies from 1100 µs, where the blade rotation speed is zero, to 1900 µs, where the blades rotate at maximum speed. In this simulation, the three clockwise blades rotate at the maximum speed and the three other blades stop. As a consequence, the other components of torques and forces present steady-state error.

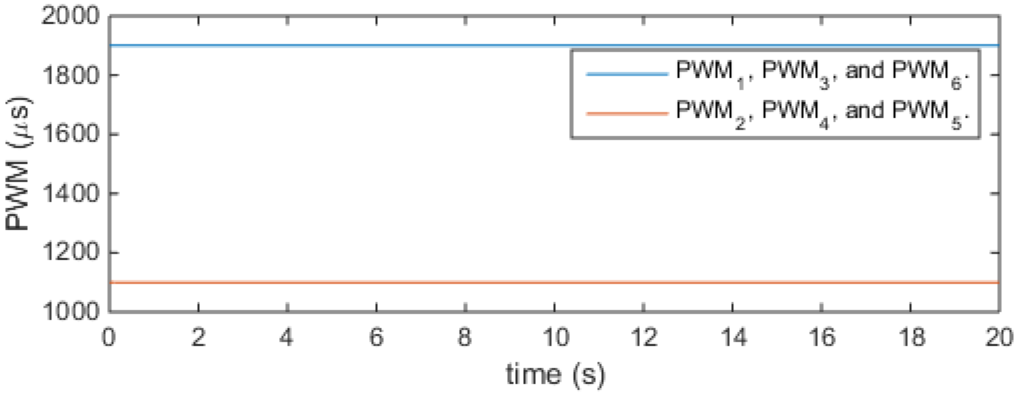


Figure 54. PWM Signals for the Lateral Force Simulation

A modified controller is simulated for the case of lateral force to study how to apply a lateral force without PWM saturation. In this simulation, the vertical torque input computed by the equilibrium-based controller is multiplied by a zero gain. Consequently, the controller does not try to compensate the torque generated by the lateral force, and the PWM signals do not saturate. In Figure 55, the forces and torques for this simulation are plotted. The lateral force $F_x$ converges to 1.0 N, the vertical interaction torque $\tau_z$ is not cancelled and the other interaction forces and torques converge to zero.

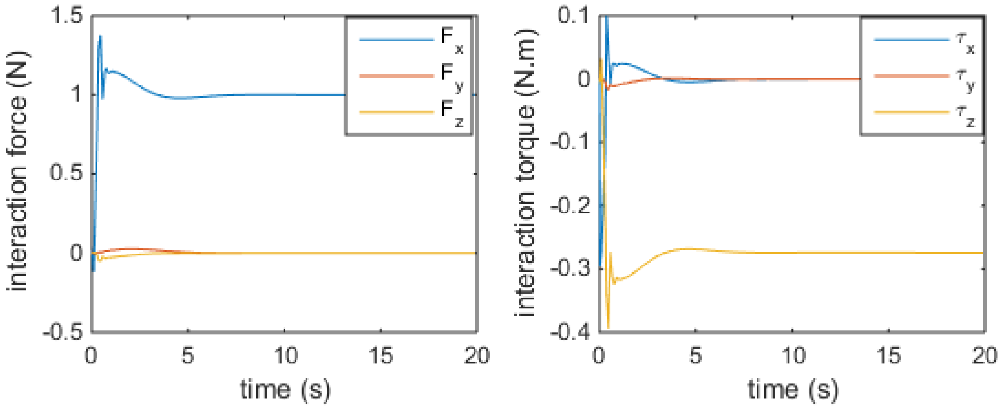


Figure 55. Forces and Torques for the Lateral Force Modified Controller

In Figure 56, the attitude roll angle by which the multicopter rotates to apply the lateral force is shown.

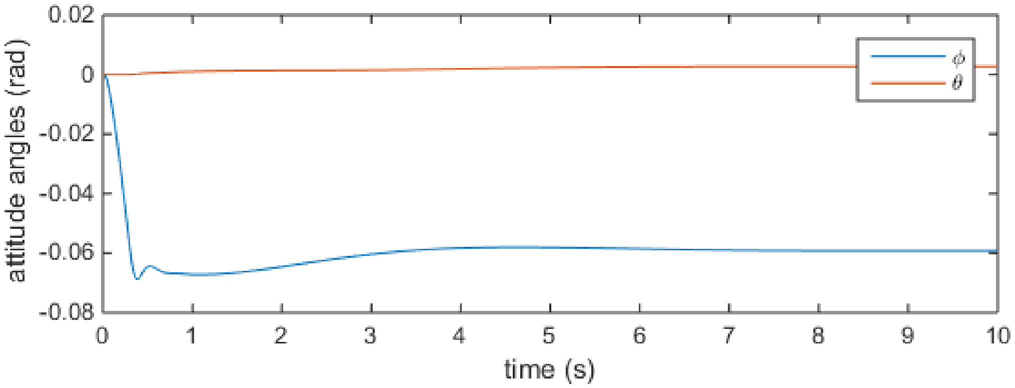


Figure 56. Attitude of the Multicopter for the Lateral Force Simulation

### 2. Longitudinal Force Interaction

The longitudinal force is implemented by making the reference force/torque $\tau_r = \begin{bmatrix} 0 & 1\text{N} & 0 & 0 & 0 & 0 \end{bmatrix}^T$. In this case, the multicopter performs a pitch movement, the thrust force is projected towards the wall, and the aerial manipulator pushes the object on the wall. In Figure 57, the interaction forces and torques are plotted. The force $F_y$ converges to 1 N, while the other components converge to zero, as expected. The pitch angle by which the multicopter rotates to apply the desired force is shown in Figure 58.

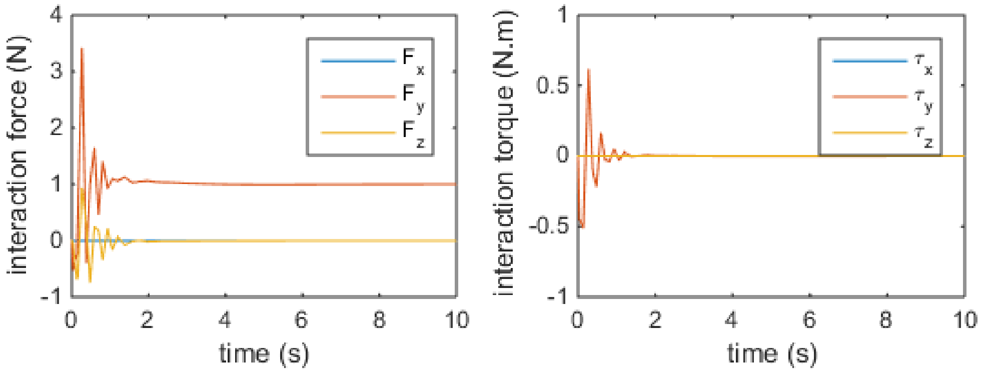


Figure 57. Interaction Forces and Torque for the Longitudinal Force Case

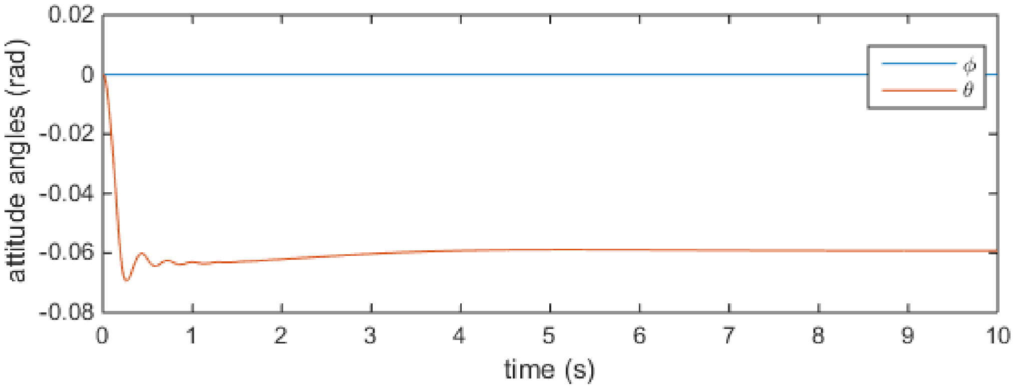


Figure 58. Attitude Angles for the Longitudinal Force Case

### 3. Vertical Force Interaction

The reference force/torque vector for the vertical force interaction is initially null, and after 7 s is set to $\tau_r = \begin{bmatrix} 0 & 0 & 1\text{N} & 0 & 0 & 0 \end{bmatrix}^T$. The aerial manipulator applies an upward vertical force by increasing the multicopter total thrust. In Figure 59, the forces and torques for the vertical force interaction are plotted. The attitude of the multicopter does not vary, which allows a fast convergence to the desired force only by changing the PWM signals.

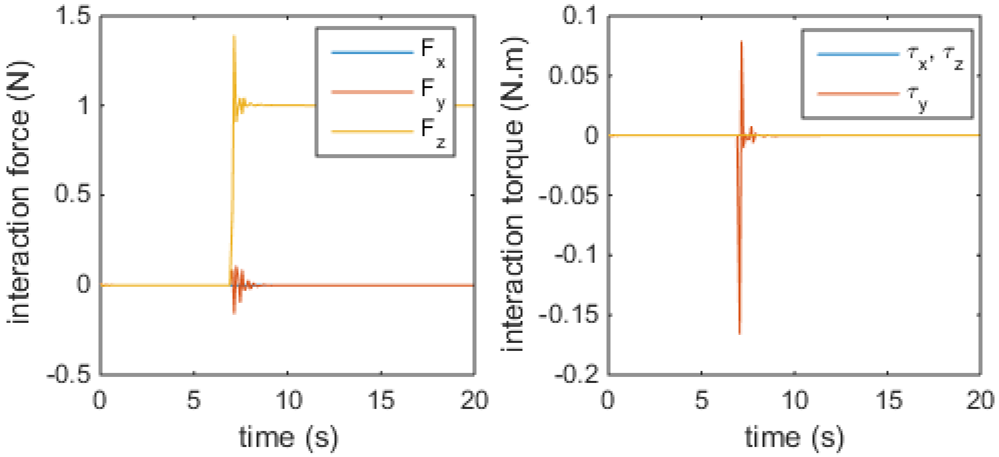


Figure 59. Interaction Forces and Torques for the Vertical Force Interaction

In Figure 60, the PWM signals increase in average. When the multicopter thrust increases, the aerial manipulator behaves as a wrench. In order to compensate the wrench torque caused by $F_z$, the PWM signals of the anterior blades $PWM_3$ and $PWM_5$ are increased, while the PWM signals of the posterior blades $PWM_4$ and $PWM_6$ are decreased.

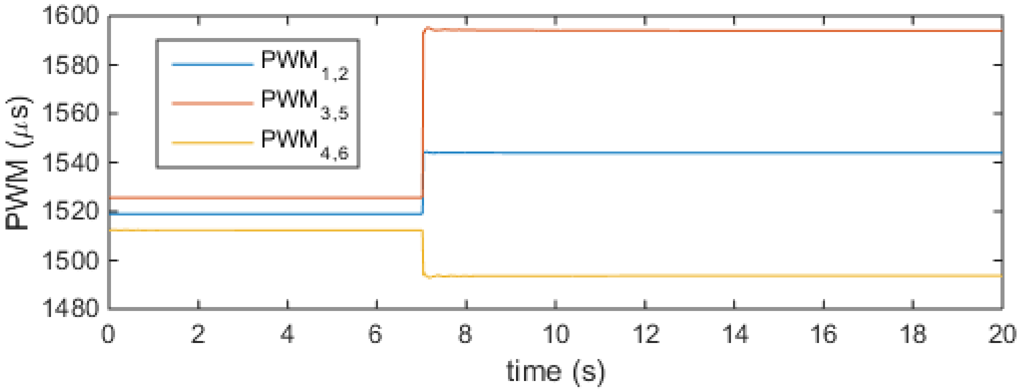


Figure 60. PWM Signals for the Vertical Force Interaction

### 4. Roll Torque Interaction

The reference force/torque vector for the roll torque simulation is given by $\tau_r = \begin{bmatrix} 0 & 0 & 0 & 1\text{N.m} & 0 & 0 \end{bmatrix}^T$. In order to apply a torque on the object on the wall, the aerial manipulator applies a roll torque on the multicopter body. This torque is induced on the object through the robotic arm. In Figure 61, the forces and torques for the roll torque interaction are plotted.

The wall external forces and torques are modeled as a damped spring system. For this reason, when the roll torque is applied on the object, the aerial manipulator rotates in compliance with the spring. After the multicopter rotates, the aerial manipulator interaction controller corrects the attitude roll angle, as shown in Figure 62. The robotic arm first joint rotates to compensate the attitude error, as in Figure 63. The interaction controller ability to correct the attitude angles is very important because the experimental robotic arm is not ideally rigid.

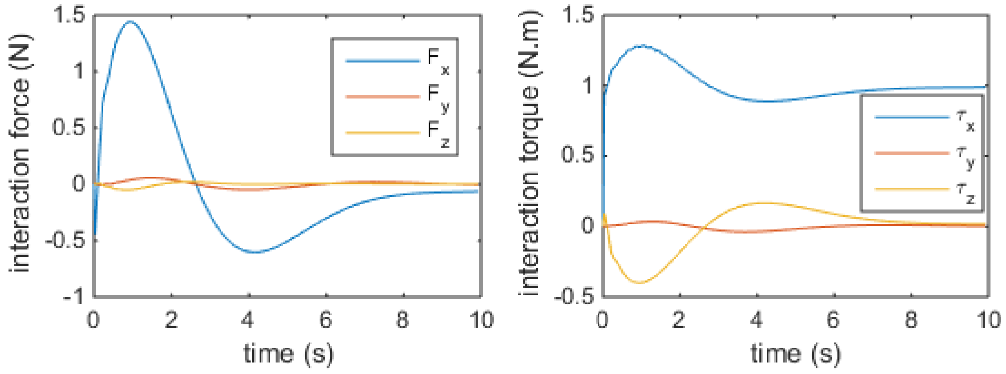


Figure 61. Forces and Torques for the Roll Torque Interaction Simulation

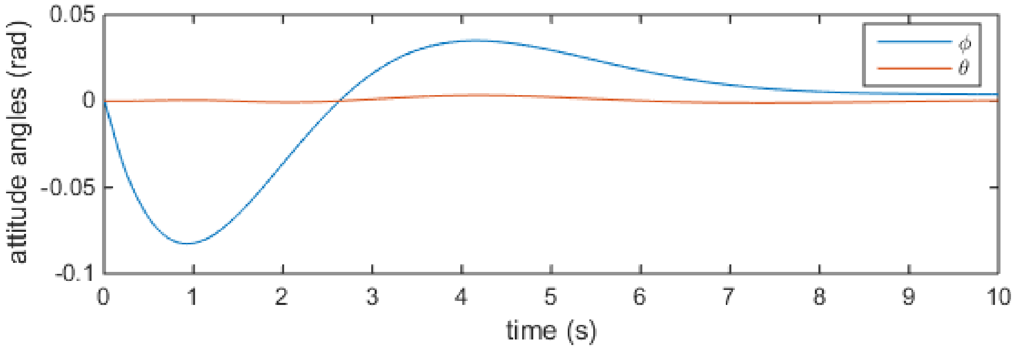


Figure 62. Attitude Angles for the Roll Torque Interaction Simulation

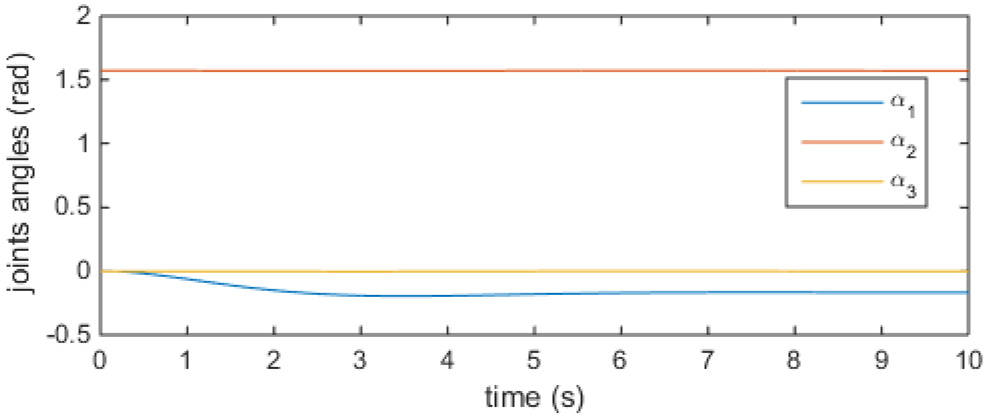


Figure 63. Robotic Arm Angles for the Roll Torque Interaction Simulation

### 5. Pitch Torque Interaction

In order to study the pitch torque interaction, the reference vector is set to $\tau_r = \begin{bmatrix} 0 & 0 & 0 & 0 & 1\text{N.m} & 0 \end{bmatrix}^T$. The aerial manipulator interaction controller applies a pitch torque on the multicopter body. This torque is induced on the object through the robotic arm. In Figure 64, the interaction forces and torques are plotted and the attitude angles are shown in Figure 65. The robotic arm corrects the attitude angles by rotating the second and third joints, as in Figure 66. While the attitude angles are corrected, the pitch torque converges to 1 N.m and the other forces and torques converge to zero, as expected.

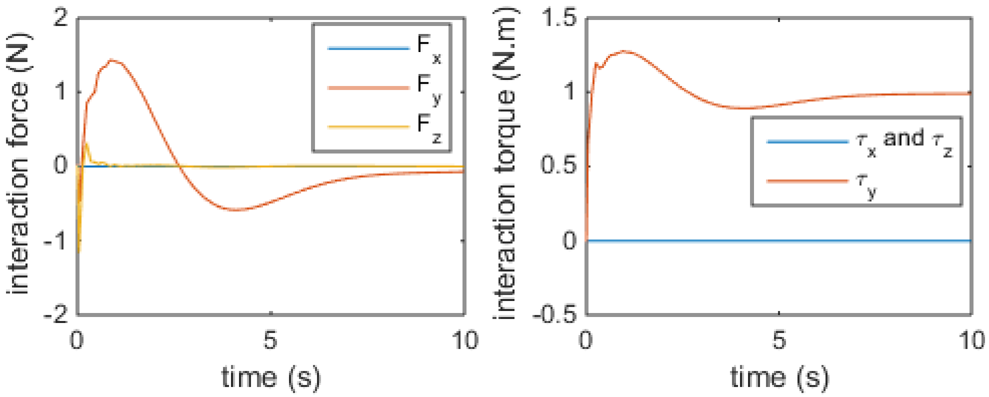


Figure 64. Forces and Torques for the Pitch Torque Interaction

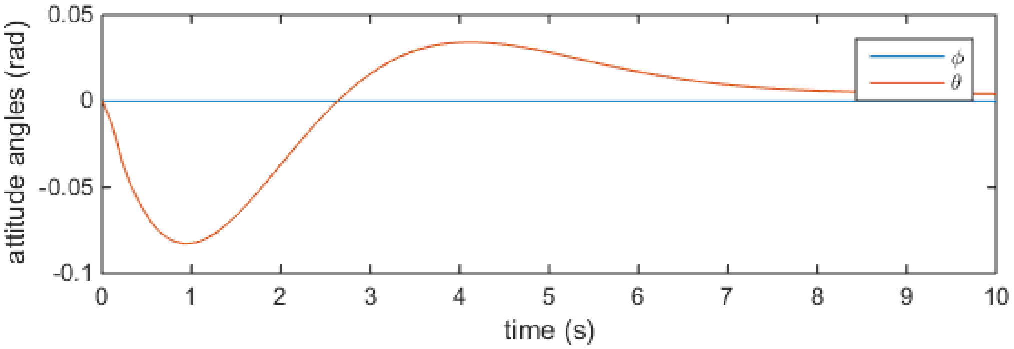


Figure 65. Attitude Angles for the Pitch Torque Interaction

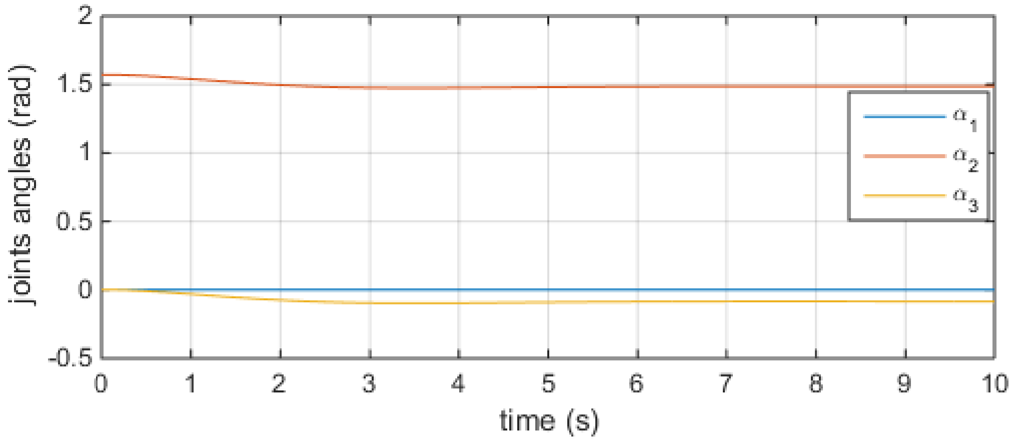


Figure 66. Joints Angles for the Pitch Torque Interaction

### 6. Yaw Torque Interaction

As already mentioned, the multicopter propellers are not able to generate large yaw torques. In order to avoid PWM saturation, the yaw reference torque is set to a small value by doing $\tau_r = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0.1\text{N.m} \end{bmatrix}^T$. Since the attitude does not vary, the system convergences considerably fast, as in Figure 67. If an aerial manipulation task requires a larger yaw torque, the aerial manipulator may be set to apply a lateral force, as already shown in Figure 55.

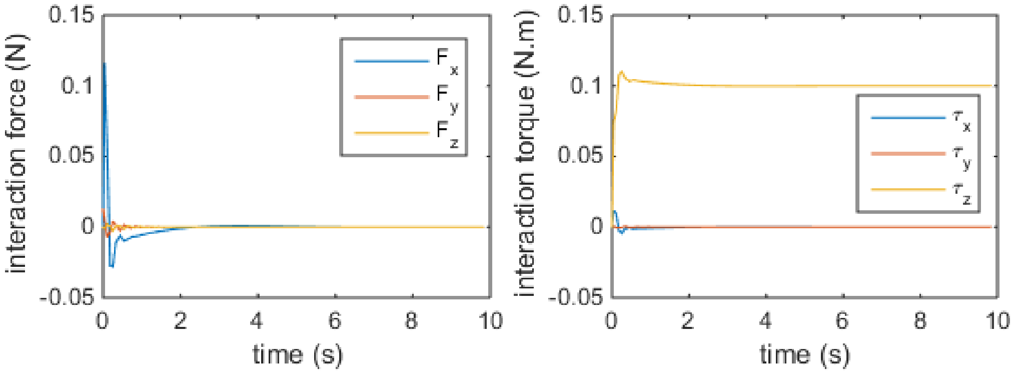


Figure 67. Forces and Torques for the Yaw Torque Interaction

### 7. Force/Torque-Combined Interaction

In this simulation case, the closed-loop controller is tested for a condition where torques and forces are applied simultaneously. The reference interaction vector is set to $\tau_r = [1\text{N} \quad 1\text{N} \quad 1\text{N} \quad 1\text{N.m} \quad 1\text{N.m} \quad -0.2\,\text{N.m}]^T$. The yaw torque value is chosen to avoid PWM saturation. All the forces and torques converge to their reference levels, as shown in Figure 68.

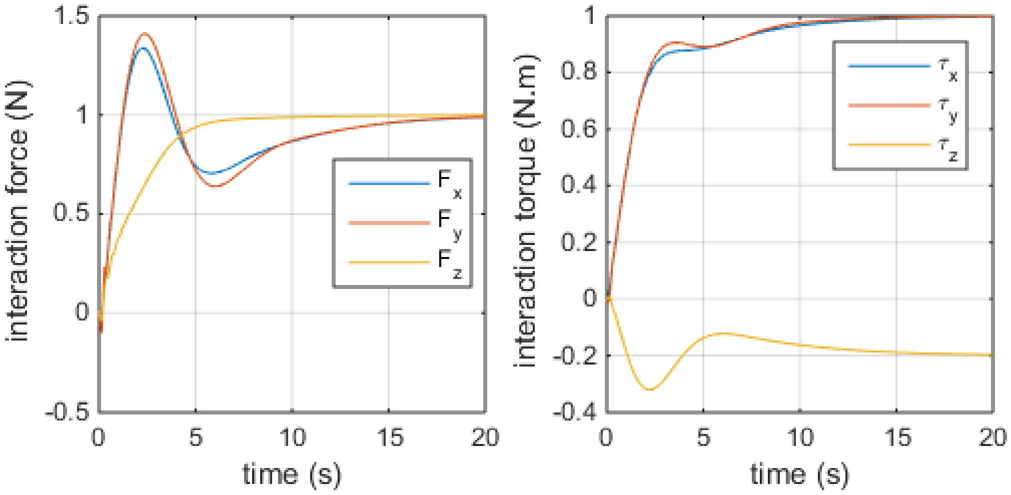


Figure 68. Forces and Torques for the Force/Torque-Combined Interaction

### 8. Open-Loop Controller Experiment

In this simulation, the force/torque sensor feedback is removed to test the controller for situations where this sensor is not available. The reference vector in this simulation is given by $\tau_r = [0 \quad 3\,\text{N} \quad 0 \quad 0 \quad 0 \quad 0]^T$. This reference force was chosen to highlight the main problem of an open-loop controller, the steady-state error. The aerial manipulator weight is approximately 17 N. In order to project 3 N along the *y* axis, the multicopter needs to rotate to a high pitch angle. In Chapter V, the equilibrium-based force-torque controller is developed with small attitude angle approximation. For this reason, some forces and torques present steady-state error, as shown in Figure 69. It is also shown in the plots that the forces and torques converge with stability. The open-loop controller is recommended for applications that do not require accurate forces and torques.

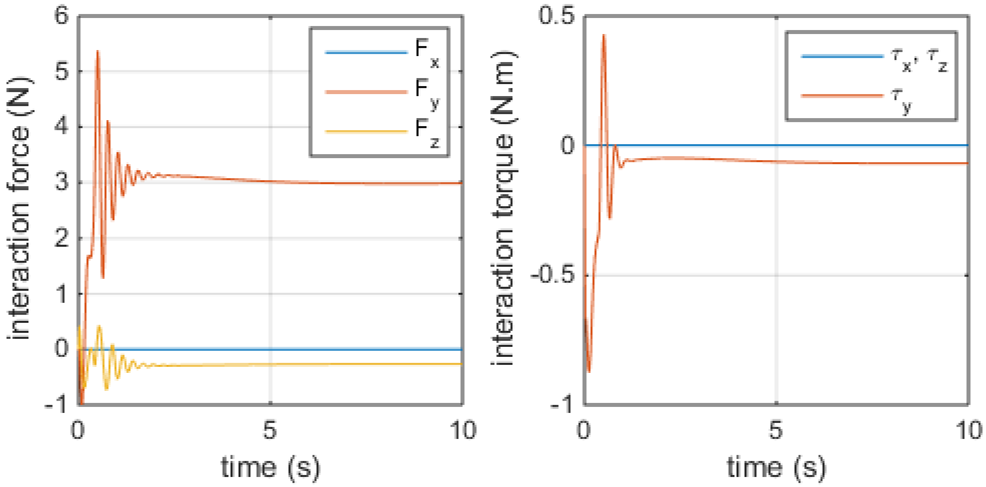


Figure 69. Forces and Torques for the Open-Loop Controller Simulation

# VII. EXPERIMENTS

In this chapter, the equipment, laboratory facilities and the experiments to validate the theoretical results are presented. A hexacopter with a microcontroller Pixhawk is described, as well as a motion capture system, a sensor to measure the force/torque interaction between a robotic arm and a wall, and a three-DoF robotic arm. It is explained how those items are integrated to build the experimental setup. During the first interaction force/torque experiments, the force/torque sensor was damaged and could not be repaired in time for the last experiments. It is explained how a spring-mounted device was used as an alternative solution for the interaction force experiments. The experiments for the evaluation of the aerial manipulator free-flight controller, the near-wall effect characterization, and validation of the equilibrium-based force/torque controller are described.

## A. EXPERIMENTAL SETUP

The research experiments were conducted with a hexacopter fabricated at the Naval Postgraduate School, shown in Figure 70. It is a small multicopter with a primary power source provided through a tether by a ground power supply. A secondary power source is provided by a backup LiPo battery. The hexacopter is equipped with T-Motor KV750 electric engines, and E-prop carbon propellers.

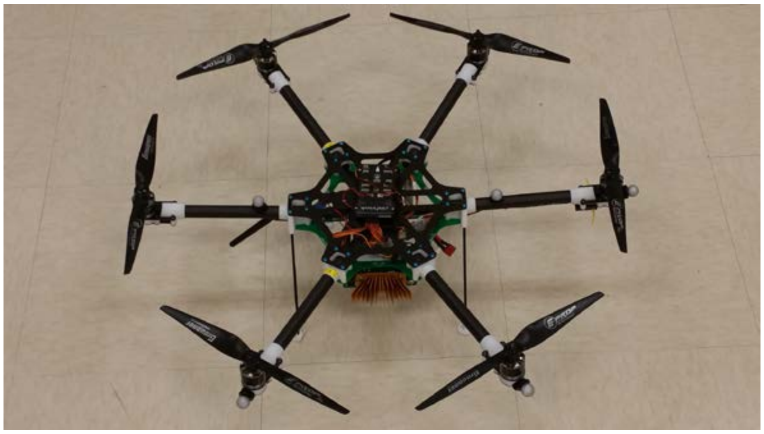


Figure 70. Hexacopter from Spacecraft Robotics Laboratory

The hexacopter is equipped with a microcontroller Pixhawk. It is an open-architecture microcontroller provided with accelerometer, gyroscope, magnetometer, and barometric altimeter sensors. It supports most of the radio controller receiver technologies. It has eight main PWM outputs, designed to be connected to propellers, and six auxiliary PWM outputs, used to access additional hardware, like a video camera. The microcontroller has several connectivity options, like UART, I2C, CAN, and USB ports. Pixhawk is also compatible with GPS modules for outdoor applications. The microcontroller processor has a 168 MHz ARM Cortex M4F CPU. There are some options of open-source firmware provided by a large community of users and developers. In this research, a Pixhawk support from Matlab Embedded Coder is used. This tool generates and compiles C code from a Simulink file and uploads the program to Pixhawk, as illustrated in Figure 71.

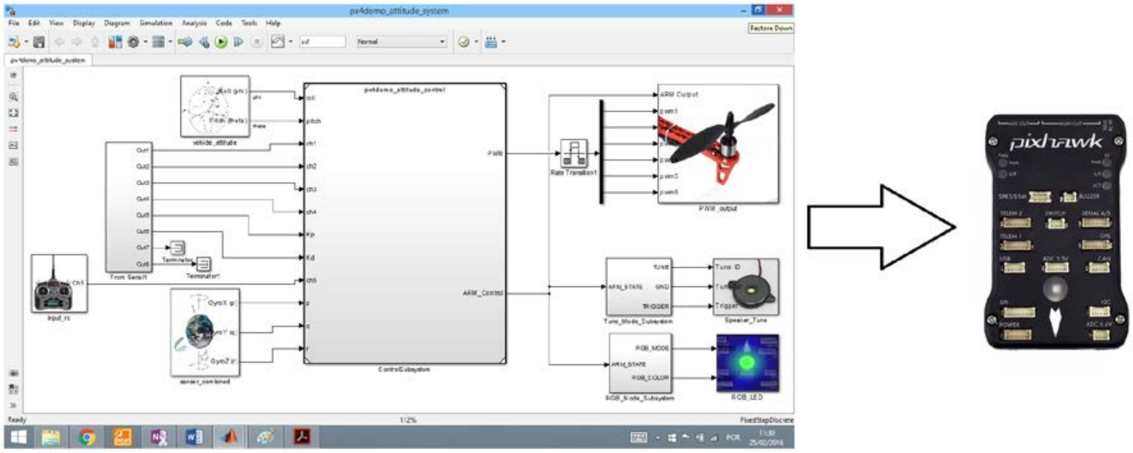


Figure 71. Simulink Controller Uploaded to Pixhawk

The aerial manipulator is equipped with a three-DoF robotic arm mounted on the bottom of the multicopter. The robotic arm has four Dynamixel AX-12A servos. Three of them rotate to control the revolute joints angles, and the last one opens and closes the end-effector. The configuration of the robotic arm is illustrated in Figure 12. The robotic arm shown in Figure 72 is controlled by an Arbotix-M board, which is an Arduino compatible device. Each Dynamixel servo has a bank of registers that are either read or written by the Arbotix-M board through a serial port. The bank of registers record the parameters associated with the motion of the servo. By writing into the registers, the controller defines parameters like the servo goal position and desired moving speed. By reading the registers, the controller obtains information related to the status of the servo, like position and velocity feedback. Another serial port connects the Arbotix-M board to the multicopter microcontroller. The Arbotix-M board is programmed to pass commands from the multicopter to the robotic arm and also send feedback from the robotic arm to the multicopter. Each servo has an internal controller to convert the motion commands into torques.

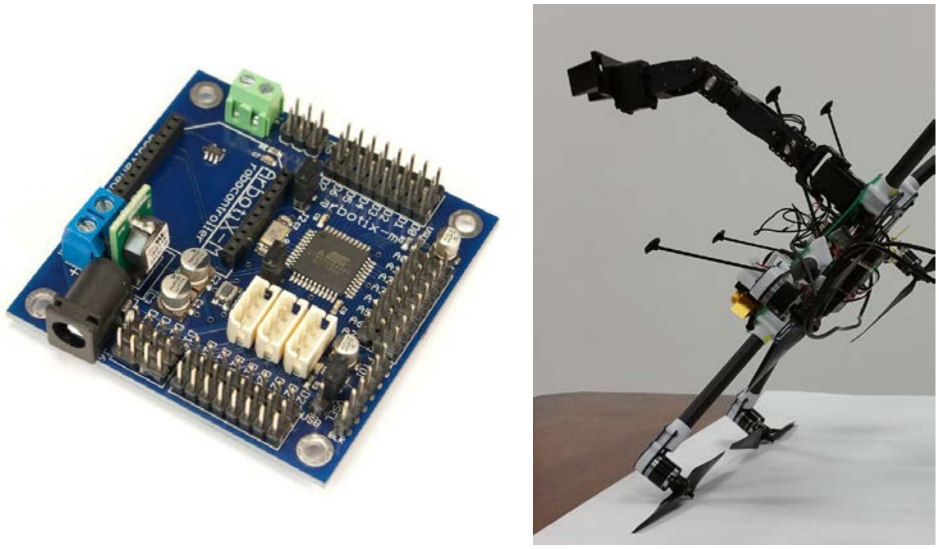


Figure 72. Arbotix-M Controller and the Robotic Arm Mounted on the Hexacopter

The Spacecraft Robotics Laboratory is equipped with a Vicon motion capture system. The laboratory testbed is surrounded by ten cameras able to detect infrared reflective markers fixed on the multicopter. The software Tracker, installed in the Vicon Server, computes the position of each marker by triangulation. Once the position of each marker is known, the software computes the position and the orientation of the multicopter.

An ATI NANO43 force/torque transducer, shown in Figure 73, measures the interaction between the end-effector and the vertical wall. The multi-axis sensor measures all the six components of force and torque. The specification of the sensor is presented in Table 3. The sensor consists of a system of strain gages that convert internal tension or compression loads into voltages. Those signals are digitalized by a National Instruments data acquisition card (DAQ) and received in Matlab environment. A calibration matrix is provided by the manufacturer to convert those voltages into forces and torques.

Table 3. ATI Nano43 Specification

| Sensing Range | | Resolution | |
|---|---|---|---|
| Force | Torque | Force | Torque |
| 18 N | 0.25 N.m | 1/256 N | 1/20000 N.m |

Figure 73. Force/Torque Sensor ATI Nano43

The experimental setup is illustrated in Figure 74. The Vicon server and a ground station are connected to the same network. A Matlab script running in the Vicon server transmits the position and the orientation of the multicopter, through UDP connection, to the ground station. The force/torque sensor is physically connected to the ground station through a data acquisition card. Part of the controller is implemented in the ground station computer, and the other part is executed in the microcontroller. The ground station and the microcontroller communicate through serial ports with XBee antenas. A serial cable connects the microcontroller Pixhawk to the robot controller Arbotix-M.

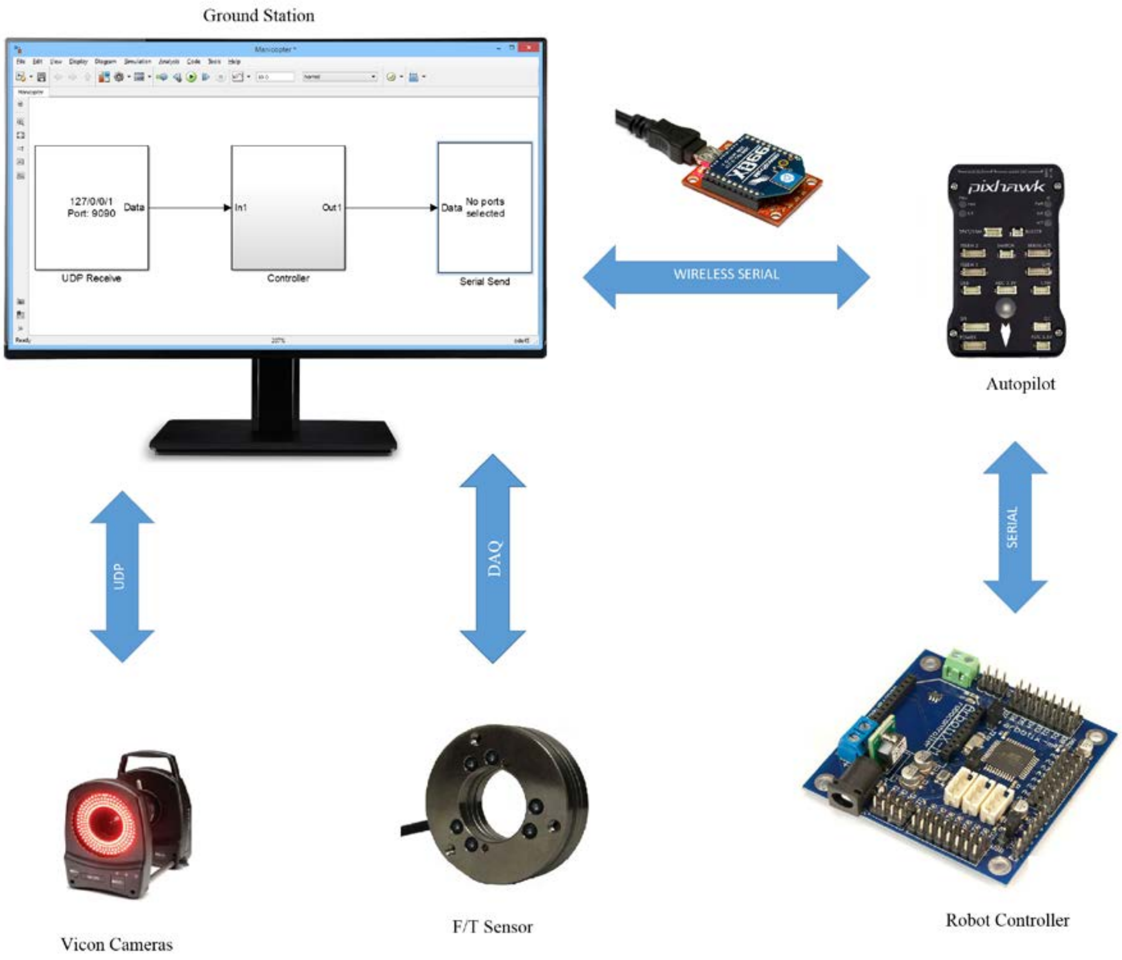


Figure 74. Experimental Setup

Some experiments to validate the equilibrium-based force/torque controller were executed with a spring-mounted device in replacement of the torque/force sensor. In Figure 75, the end-effector holds an object that slides along a rail. By pushing the sliding object, the robotic arm compresses the spring towards the fixed object on the left side. The reflective markers are detected by the Vicon cameras and the position of the sliding device is sent through UDP communication to the ground station. The compression of the spring is directly related to the position of the sliding object. The force applied on the spring is obtained by multiplying the spring constant by the spring compression. With this experimental setup, the ground station controller has feedback of the force applied on the spring by the aerial manipulator.

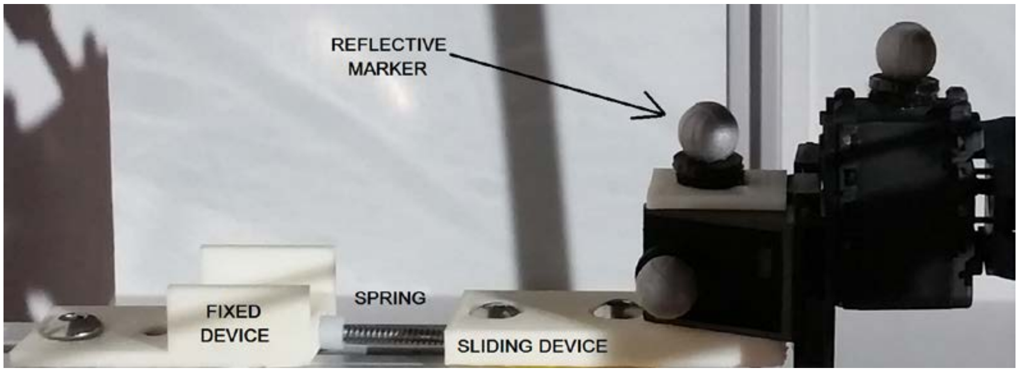


Figure 75. Spring-Mounted Device for Interaction Force Experiments

## B. FREE-FLIGHT CONTROLLER EXPERIMENTAL RESULTS

In this section, the free-flight controller experiment results are presented. In the free-flight experiment, the aerial manipulator hovers in the center of the testbed while moving the robotic arm. In Figure 76, the aerial manipulator is shown for the different robotic arm positions. The sequence of robotic arm movements is detailed in Figure 77. The multicopter hovers for 60 s for each robotic arm position.

The attitude control law is implemented by defining the attitude torque $\tau_{\Phi}$ as in Equation (60). The column vector $G(q)$ in the control law equation represents the effect of the gravity on each generalized coordinate. The fourth and fifth elements of $G(q)$ are associated to the attitude angles $\phi$ and $\theta$, respectively. These two elements of the gravity vector represent the roll and pitch torques applied by the multicopter to compensate the gravity. When the robotic arm moves, the attitude controller computes the new values of $G(q)$ and the gravity is compensated instantaneously. There are other factors that affect the performance of the attitude controller during free flight. During the experiments it was observed that if the robotic arm links are very close to a blade, the lift force of that blade is affected. Since this aerodynamic effect is not modeled, the integrator term in the attitude controller converges to the torque necessary to compensate this effect.

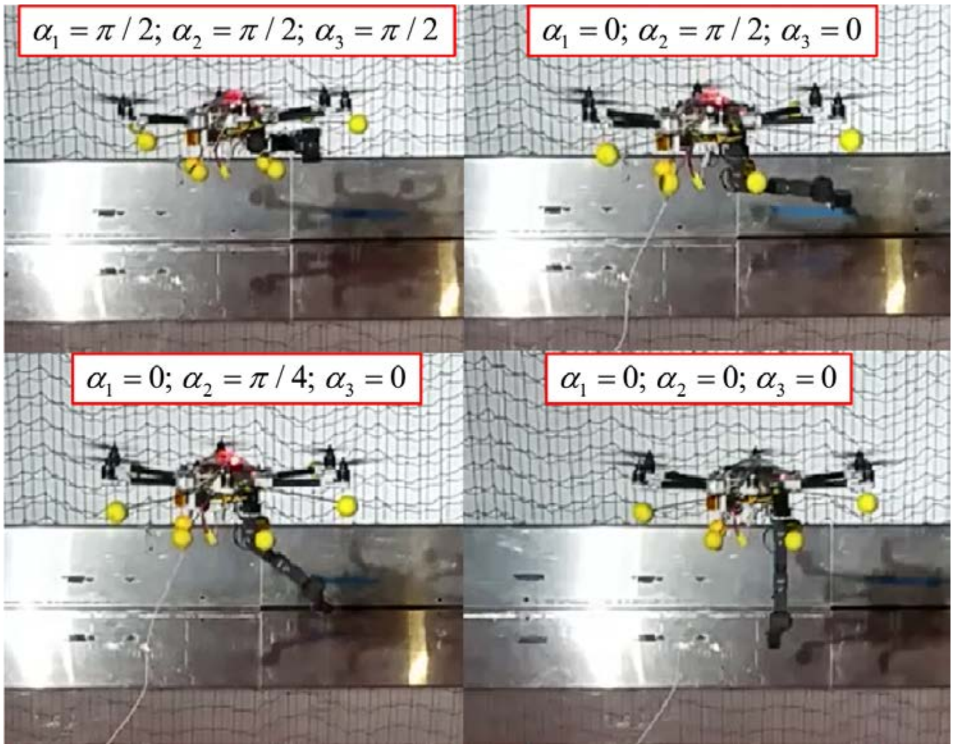

$\alpha_1 = \pi/2; \alpha_2 = \pi/2; \alpha_3 = \pi/2$
$\alpha_1 = 0; \alpha_2 = \pi/2; \alpha_3 = 0$
$\alpha_1 = 0; \alpha_2 = \pi/4; \alpha_3 = 0$
$\alpha_1 = 0; \alpha_2 = 0; \alpha_3 = 0$

Figure 76. Aerial Manipulator Moving the Robotic Arm

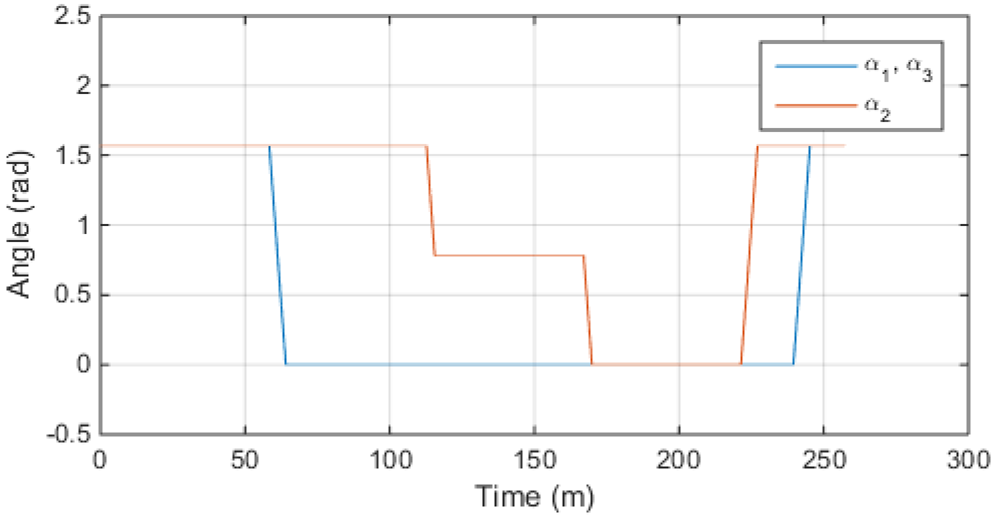


Figure 77. Joints Angles for the Free-Flight Experiment

In Figure 78, the output torque of the integrator and the gravity compensation torque applied by the attitude controller are plotted. Initially, the robotic arm is very close to the hexacopter blades. During the transition from $\alpha^T = [\pi/2 \quad \pi/2 \quad \pi/2]$ to $\alpha^T = [0 \quad \pi/2 \quad 0]$, the robotic arm moves away from the hexacopter. The integrator roll torque varies significantly because robotic arm aerodynamic disturbance is not relevant after the transition. The gravity torque is the most relevant factor during the following transitions. The integrator torques do not vary significantly, indicating that the gravity torque compensation works effectively. After the last transition, the robotic arm moves back to the initial position and the integrator roll torque varies to compensate the robotic arm aerodynamic disturbance.

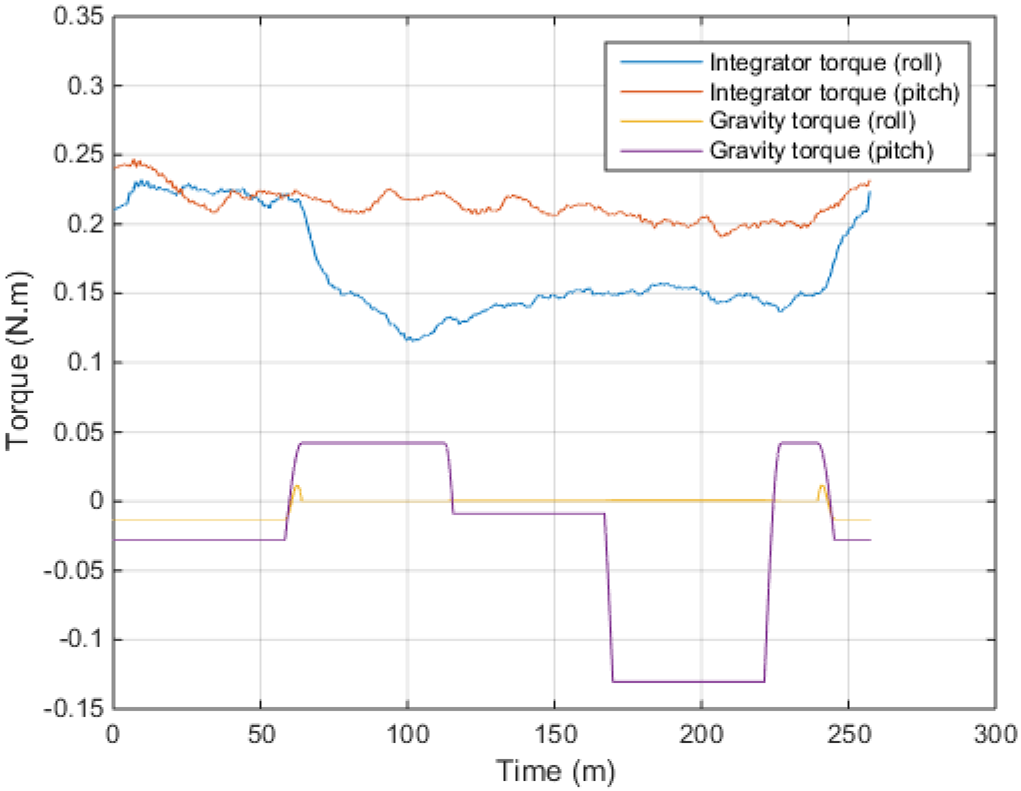


Figure 78. Integrator and Gravity Compensation Torques during Flight Experiment

In Figure 79, the position of the aerial manipulator during the free-flight experiment is plotted. The aerial manipulator hovers at 60 cm from the ground and moves the arm to the next arm position every 60 s. The multicopter position remains relatively constant even when the robotic arm is moving. The multicopter testbed is not very large. When the hexacopter turns on the rotors, the air circulation in the room affects the flight performance especially in the horizontal plane. The position standard deviations for this experiment are 10.38 cm in the *x* direction, 10.33 cm in the *y* direction, and 3.55 cm in the *z* direction.

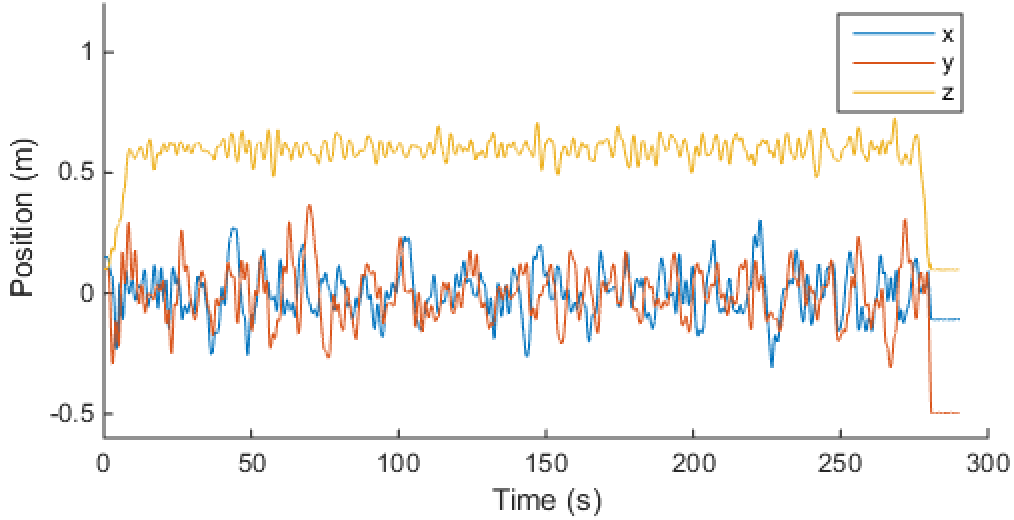


Figure 79. Multicopter Position during Free-Flight Experiment

## C. NEAR-WALL EFFECT CHARACTERIZATION

In this section, the experiment for the characterization of the near-wall effect is presented. Next, a near-wall effect compensator is proposed to be integrated to the attitude controller as illustrated in Figure 27.

In Figure 80, the experimental setup for the near-wall effect measurement is shown. The aerial manipulator is suspended by a line fixed to a cage. The thrust force is set to a constant force $F_t = 10\,\mathrm{N}$, smaller than the aerial manipulator weight. This configuration allows the multicopter to rotate freely to any direction, at a fixed position. The wall can be safely placed near the blades while the rotors are turned on. When the multicopter is turned on, the attitude controller integrator converges to a torque to compensate the steady-state attitude errors. In this experiment, the near-wall effect is the main cause of steady-state attitude error. The first experiment is performed to measure the torque without wall. The integrator attitude torque associated to the pitch movement is plotted in Figure 81.  The integrator converges before the last 30 seconds of experiment. The same measurement was performed with the wall placed in front of the hexacopter at different distances separating the wall and the closest blades. For each experiment, the integrator average torque for the last 30 seconds is calculated. The steady-state average pitch torque for each experiment is presented in Figure 82.

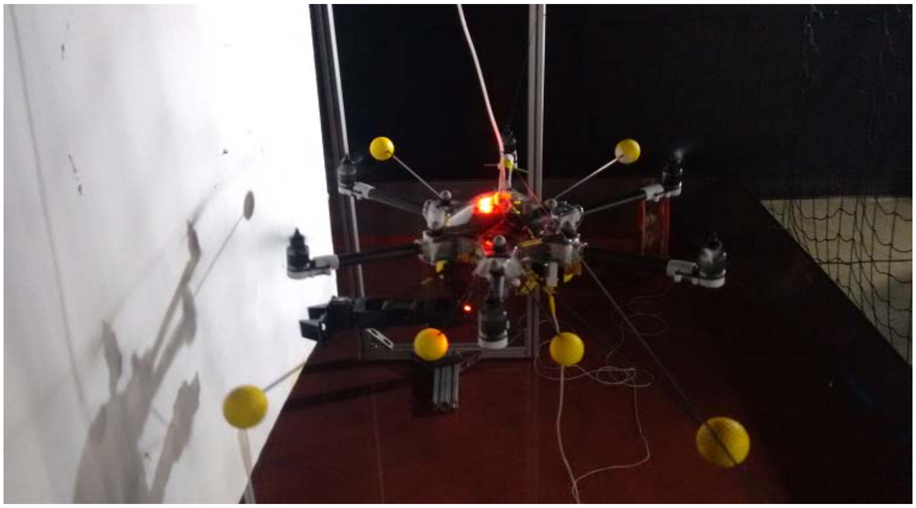


Figure 80. Near-Wall Effect Characterization Experiment

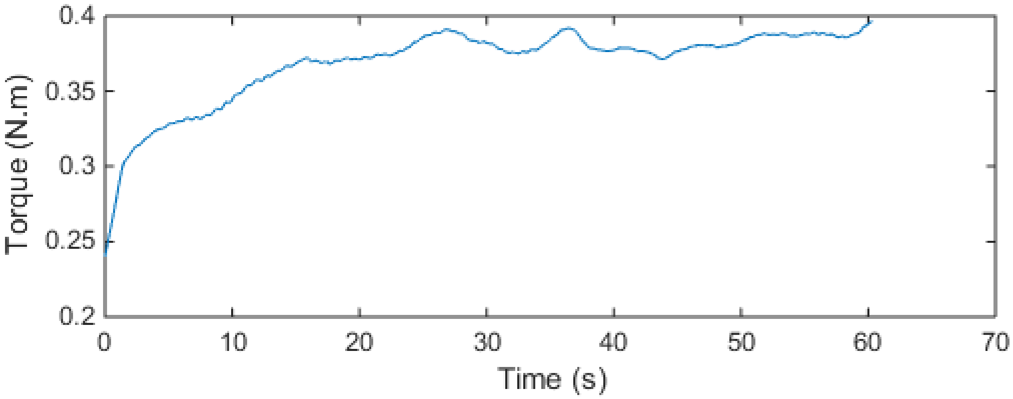


Figure 81. Integrator Pitch Torque for the No-Wall Case

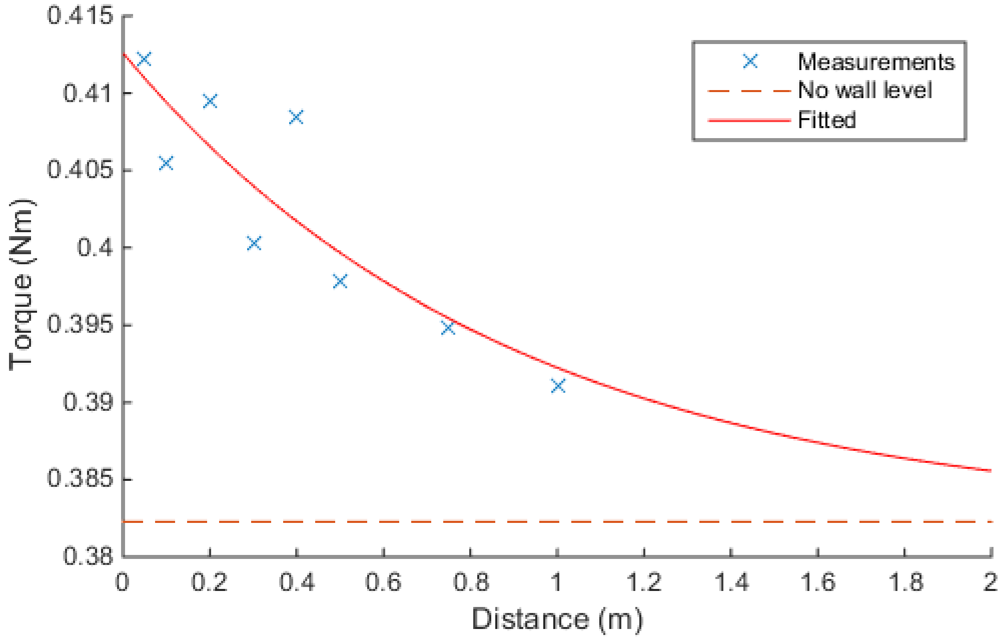


Figure 82. Near-Wall Experiment for 10 N of Thrust

An exponential curve is fitted to the experimental data. The estimated near-wall torque is a function of the distance *d* separating the blades and the wall, given by

$$\tau_{nw} = 0.030e^{-1.113d} \ . \tag{111}$$

The experiments were performed with a fixed thrust $F_t = 10\mathrm{N}$, but the aerial manipulator weight is 16.84 N. With a smaller thrust force, the aerial manipulator could be suspended by the line. In the near-wall effect study described in [30], it is shown that the near-wall effect on a rotorcraft is proportional to the vehicle weight. For this reason, the near-wall torque in Equation (111) is modified. The near-wall torque compensator for the experimental aerial vehicle is given by

$$\tau_{nw} = 0.051e^{-1.113d} \ . \tag{112}$$

This curve is proposed as the near-wall torque compensator, to be included in the attitude control law in Figure 27.

## D. WALL INTERACTION EXPERIMENTAL RESULTS

In this section, the experiments to validate the equilibrium-based force/torque controller are presented. The first experiments were performed with a six-dimensional force/torque sensor. The sensor got damaged and could not be repaired in time for the last experiments. The final experiments were performed without the force/torque sensor. One experiment was designed with the spring system already described to prove that the controller is able to regulate a specified force on the object. Other experiments were performed without force or torque feedback to validate the controller when the aerial manipulator is required to apply a specified torque with stability.

In Figure 83, the aerial manipulator is suspended by a line while grasping the force/torque sensor. The first experiments were performed with small thrust forces. By having the thrust force smaller than the aerial manipulator weight, the experimental configuration was similar to a pendulum. This approach is safer because, like a pendulum, this configuration is naturally stable. The methodology was to make tests increasing the thrust force progressively while observing the behavior of the aerial manipulator to tune the controller gains to achieve stability. In the end of this process, the aerial manipulator would be able to fly by itself during force/torque interaction. However, the sensor got damaged before the aerial manipulator could fly without the assistance of the suspension line. In Figure 84, the sensor feedback during one of these experiments is plotted. The objective of the experiment was to apply a lateral force $F_x = 0.5\,\mathrm{N}$ with a thrust force $F_t = 7\,\mathrm{N}$.

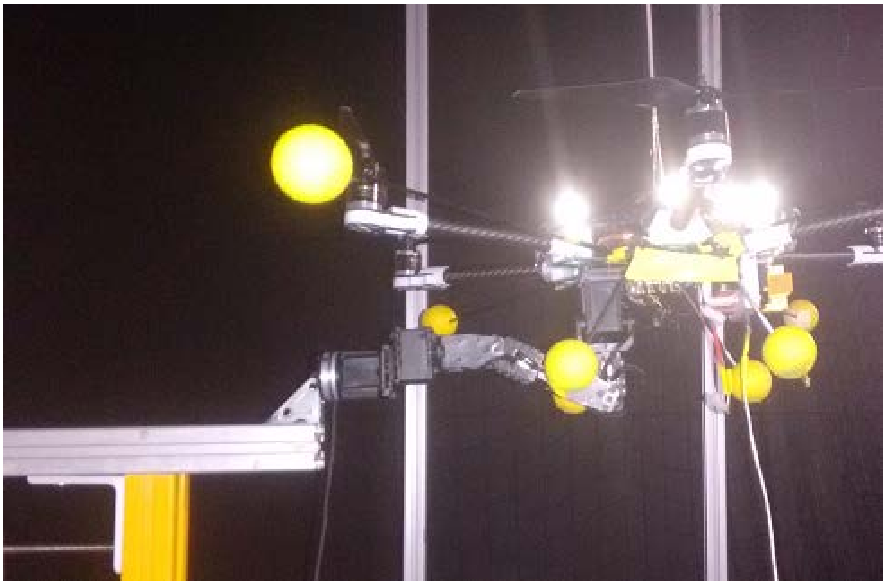


Figure 83. Force/Torque Experiment with Force/Torque Sensor

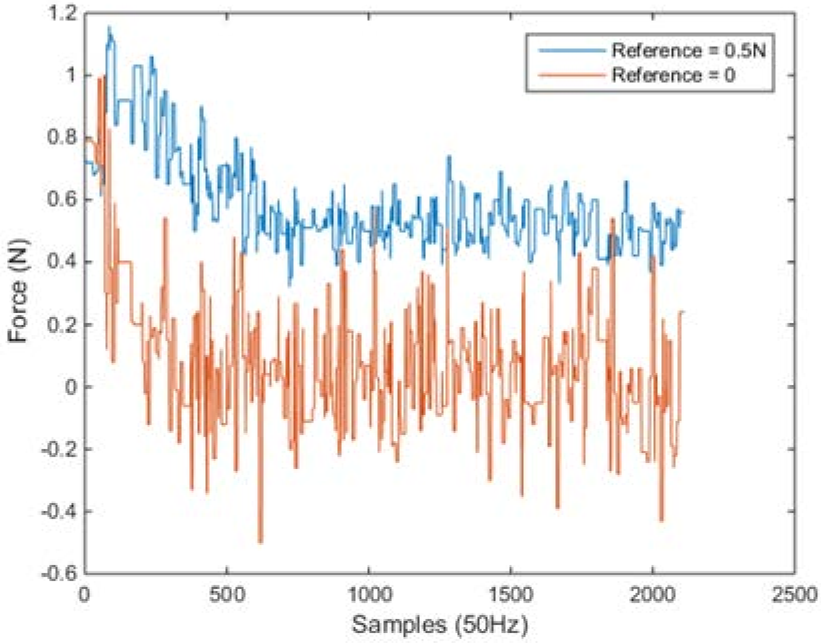


Figure 84. Longitudinal (Red) and Lateral (Blue) Forces for $F_t = 7$ N

The experiments on force/torque interaction were continued with the spring-mounted device. The same methodology previously described for the force/torque sensor experiments was followed. The experiments were conducted by progressively increasing the thrust force until the multicopter could fly without the suspension line. In Figure 85, the aerial manipulator flies while applying a longitudinal force towards the wall.

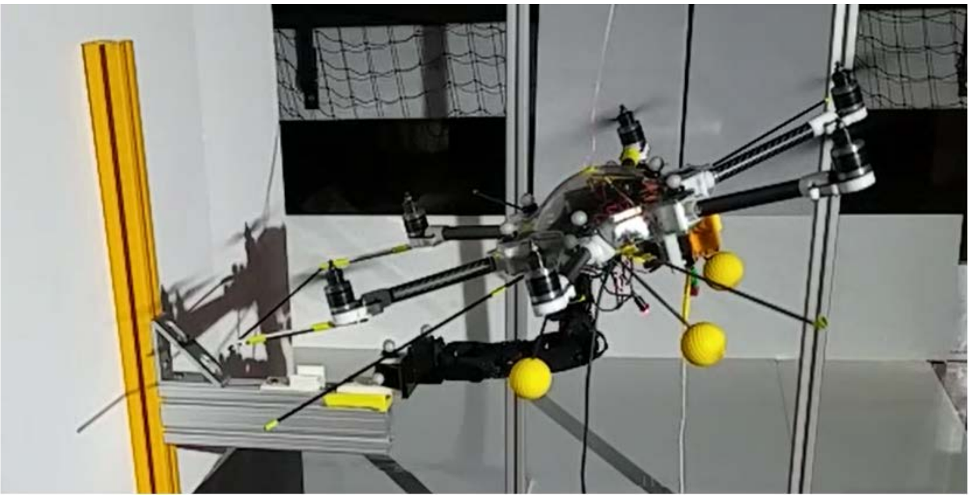


Figure 85. Aerial Manipulator Force/Torque Interaction Experiment

Several experiments were performed to study the force interaction. The aerial manipulator objective during the experiments with the spring-mounted device was to control the force applied on the spring. In Figure 86, Figure 87, and Figure 88, the forces applied on the spring for three different reference forces are plotted. In all the experiments, the equilibrium-based force/torque controller successfully regulated the forces on the spring. It was observed significant friction between the sliding object and the rail. In some experiments the aerial manipulator needed more time to converge and maintain the desired force, but the convergence was always achieved.

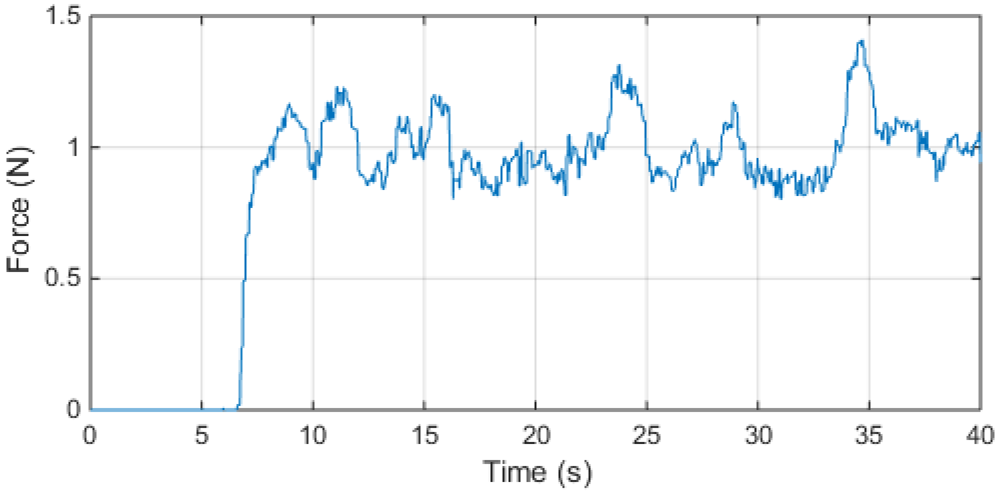


Figure 86. Interaction Force Experiment for a Reference Force $F_y = 1\,\text{N}$

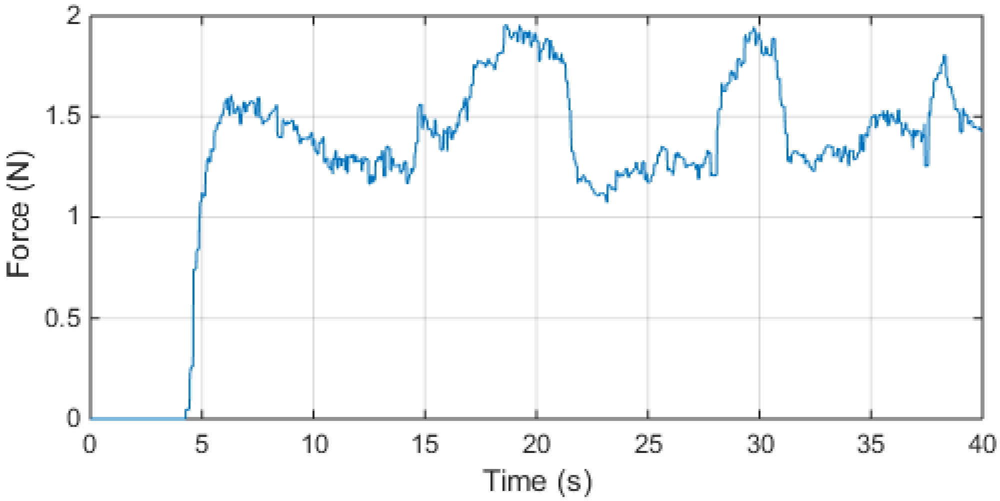


Figure 87. Interaction Force Experiment for a Reference Force $F_y = 1.5\,\text{N}$

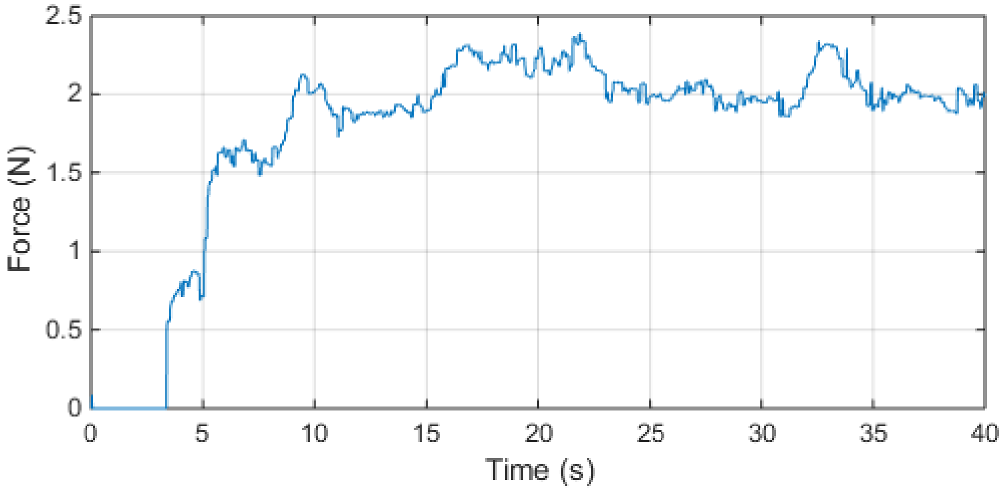


Figure 88. Interaction Force Experiment for a Reference Force $F_y = 2\text{N}$

In Figure 89, the attitude pitch angle is plotted for the same experiment in Figure 88. The projection of the thrust force on the horizontal plane towards the wall is determined by the pitch angle. For a negative pitch angle, the multicopter moves the nose down and the aerial manipulator pushes the wall. By comparing the force plot with the attitude plot, the correlation between the attitude and the interaction force is perceived.

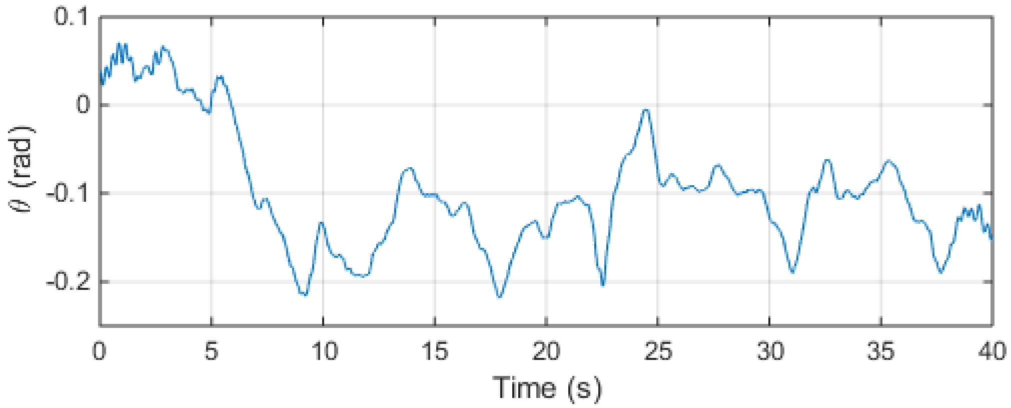


Figure 89. Hexacopter Pitch Angle for the Experiment with Reference $F_y = 2\text{N}$

Additional experiments were performed to validate the equilibrium-based force torque controller for torque interactions. Although the experiments are not configured with any kind of torque feedback, the recorded data could be analyzed to prove that the aerial manipulator is capable of applying torques with stability. In Figure 90 and in Figure 91, experiments for pitch torque and roll torque are plotted, respectively. In both experiments, the reference torque are set to be zero in the first 20 seconds approximately. After that, the reference torque is modified. When the reference torque changes, the PWM inputs also change, indicating that an additional torque is applied by the rotors and transmitted to the object through the robotic arm. In the plot in Figure 91, the multicopter oscillates during the transition and the PWM inputs are also modified by the attitude controller to damp those oscillations.

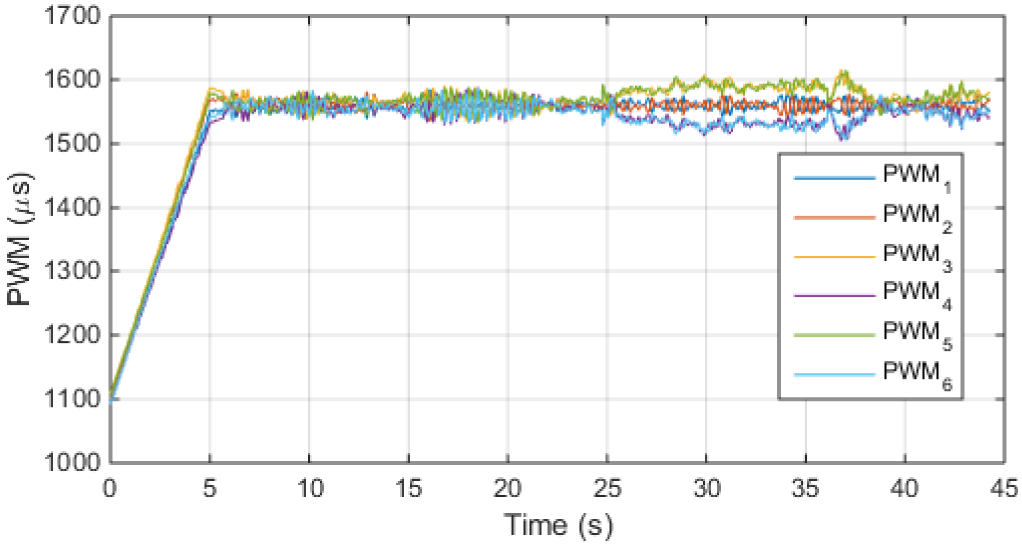


Figure 90. Interaction Torque Experiment for a Reference Torque $\tau_y = 0.5$ N.m

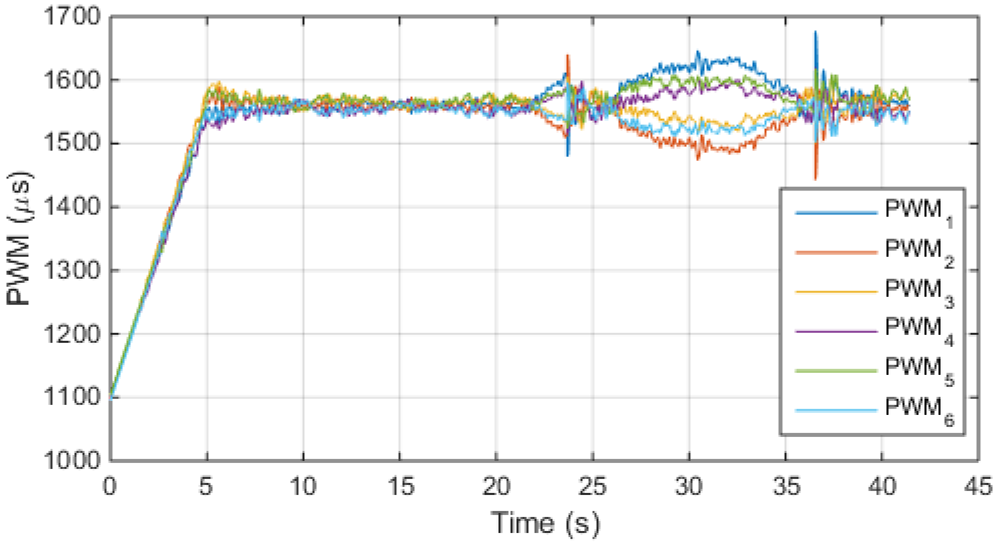


Figure 91. Interaction Torque Experiment for a Reference Torque $\tau_x = 1$ N.m

# VIII. CONCLUSION

In this research, the feasibility of employing a multicopter UAV equipped with a robotic arm for manipulation tasks on vertical surfaces was studied. The main objectives of this research were to characterize the near-wall effect and to develop a force/torque controller for wall interaction. In this chapter, a summary of the methodology, the contributions, and recommendations for future work are presented.

## A. SUMMARY

A mathematical model for a multicopoter with an n-link robotic arm was derived through Lagrangian formalism by following the methodology described in [1]. The mathematical model was applied to the experimental hexacopter equipped with a three-DOF robotic arm. In [33], a methodology to obtain the physical and electrical parameters of multicopters is described. The experiments reported in [33] were also conducted in the Spacecraft Robotics Laboratory with the same hexacopter used in this dissertation. In the paper, the authors apply the methodology to measure the parameters of the hexacopter. Part of these results were used in the dissertation research, and additional experiments were performed to determine the other parameters of the aerial manipulator.

In an aerial manipulator, the dynamics of the robotic arm and the multicopter are coupled. For this reason, an aerial manipulator controller offers a better performance if the system is modeled and controlled as a whole. A free-flight controller was implemented based on a global linearization method, as described in [3]. The control law considers the dynamics of the robotic arm to control the multicopter attitude. The effect of the robotic arm weight, velocities, and accelerations on the multicopter body are compensated by the controller.

An equilibrium-based force/torque controller was developed. The controller regulates the forces and torques that the aerial manipulator applies on an object fixed on a wall. The equilibrium-based force/torque controller computes the multicopter attitude and rotors inputs to induce the desired force and torque on the wall. With the multicopter inertial sensors feedback, the robotic arm works in collaboration with the rotors to control the attitude of the multicopter. A force/torque sensor feedback is integrated to the scheme to eliminate steady-state errors.

Simulations were performed to evaluate the free-flight controller. In the free-flight controller simulations, the aerial manipulator controller was compared to a multicopter controller without robotic arm compensation. The performance of the aerial manipulator controller is significantly superior because the coupled dynamics of the robotic arm and the multicopter are taken into consideration in the control law.

The aerial manipulator free-flight controller requires large processing load. The experiments are configured with the aerial manipulator moving at relatively low speed. For this reason, some control law terms associated with high-speed maneuvers were removed in simulations to study their significance. The centrifugal and Coriolis forces as well as the angular accelerations of the robotic arm joints were removed from the control law. This simplified version of the free-flight controller produced similar performance when compared to the original aerial manipulator controller. The simplified version of the free-flight controller was implemented in the experiments.

Simulations were performed to study the equilibrium-based force/torque controller. The regulation of each component of the six-dimensional force/torque vector was studied. The force/torque controller successfully regulated torques and forces in the six simulation cases. Simulations also validated the controller for a manipulation task with forces and torques applied simultaneously in all directions. Another simulation addressed the efficacy of the force/torque controller without force/torque sensor feedback. The controller effectively regulated the forces and torques, but a steady-state error was observed.

Experiments were performed to analyze the aerial manipulator free-flight controller. The attitude controller compensates efficiently the effect of the gravity forces on the multicopter body for each robotic arm position. It was observed that when the links of the robotic arm are moved close to hexacopter blades, the thrust generated by these blades is affected. The integrator term in the attitude controller compensates this aerodynamic effect.

The near-wall effect for single-blade helicopters is characterized through a computational method in [30]. In this dissertation research, the near-wall effect for multicopters was analyzed starting from the single-blade study. When the experimental hexacopter hovers near a wall, a pitch torque is induced on the body. Experiments were performed to measure the near-wall torque for varying distances separating the multicopter and the wall. An exponential curve was fitted to the experimental data to estimate the near-wall pitch torque as a function of the distance. A modification on the attitude controller is proposed to include a near-wall compensator in the control law.

The equilibrium-based force/torque controller was developed for a manipulator with force/torque sensor feedback. During the experiments to validate the equilibrium-based force/torque controller, the force/torque sensor was damaged and could not be repaired in time for the final experiments. For this reason, the final experiments were carried out without force/torque sensor feedback. Some experimental results show that the aerial manipulator is capable of applying torques on the wall with stability, although the accurate value of the torques cannot be measured. A spring-mounted device was designed to validate the use of the force/torque controller for force regulation. Controlling a spring force in only one direction may seem easier than controlling simultaneously a six-dimensional force torque vector with a force/torque sensor; however, the spring device was mounted on a rail with a high amount of friction. The experimental results show that the force/torque controller is robust and effectively regulates the force applied on the spring while flying with stability.

## B. CONTRIBUTIONS

One of the main contributions of this dissertation is the characterization of the near-wall effect. The current theory on single-blade helicopters was the basis for the investigation of the near-wall effect on multicopters. With a novel experimental setup, the near-wall effect was safely measured at very close distances from the wall. A near-wall compensator was proposed to improve the flight performance of a multicopter flying near a vertical surface. With the compensator, the multicopter reacts to wall disturbances instantaneously.

Another main contribution of this dissertation is the development of the equilibrium-based force/torque controller. Several experiments and simulations were performed to validate the controller for the experimental aerial manipulator used in this research. The controller was proven to be robust when applying a regulated force on a spring-mounted setup with high level of friction. The controller was developed for a generic multicopter equipped with a multi-link robotic arm to apply any force/torque combination on an object near a vertical wall. For this reason, the equilibrium-based force/torque controller has potential to be used in large range of applications.

The contributions of this study are not limited the near-wall characterization for multicopters and the development of the equilibrium-based force/torque controller. A Simulink model was developed to study the experimental aerial manipulator. The derivation of the dynamic model of the aerial manipulator requires the computation of partial derivatives of energy functions. An efficient way to compute those partial derivatives recursively was developed. A virtual environment was developed to visualize the simulations in 3D animation. The virtual environment is a helpful tool to analyze simulation results. It is also possible to reproduce experiments in 3D animation from recorded experimental data.

The experimental setup was fully integrated for this research. A few months were spent on investigating a reliable way to send commands from the ground station to the microcontroller without delay. Some options of open-source firmware were tested to define the Mathwork support package for Pixhawk as the most appropriate tool to upload firmware to the microcontroller. The Vicon system was integrated to the multicopter ground station computer by the UDP communication protocol. The force/torque sensor was integrated to the ground station computer, with force/torque feedback accessible in Matlalab. The Arduino code running in the Arbotix-M robot controller was developed to implement the serial port communication between the robot controller and Pixhawk. Currently in the Spacecraft Robotics Laboratory, other experiments on multicopter research have been using the setup tools developed in this research.

## C. FUTURE WORK AND RECOMMENDATIONS

The multicopter research was recently initiated in the Spacecraft Robotics Laboratory. In the last two years, much progress has been achieved; however, there is still room for improvement on the hexacopter flight performance. With a better set of propellers and blades, the hexacopter may become more responsive to the flight commands. Other techniques and controllers may be tested to improve the position accuracy. During the flight experiments, it was observed that the hexacopter rotors generate air circulation in the room. The performance of the position controller is affected by the wind. Some ideas from the literature review are recommended as future work to improve the position accuracy with wind disturbance. For instance, in [15], a momentum-based external-forces estimator is used to estimate wind disturbances. In [18], a sliding-mode adaptive controller is used to control an aerial manipulator position with better accuracy.

The near-wall effect was characterized for a hexacopter approaching a wall orthogonally, in a symmetric configuration. For this reason, the near-wall torque was restricted to the component associated with the pitch rotation. The near-wall torque is a function of the distance separating the multicopter and the wall and the approaching yaw angle of the multicopter relative to the wall. It is recommended to make experiments for different approaching angles. The pitch and the roll torques must be characterized. The near-wall effect was characterized experimentally for a hexacopter approaching an infinite wall. It is also recommended to study the near-wall effect when the multicopter is flying near a corner of two walls or near a wall with an opening like a window.

In order to execute the manipulation task, the aerial manipulator has to take off, fly towards the wall, and grasp an object fixed on the wall. The trajectory of the end-effector has to be controlled with accuracy. For this reason, studies on inverse kinematics are recommended to map the end-effector trajectory into generalized coordinates trajectories. A collision detection algorithm is also recommended to allow the aerial manipulator to switch from free-flight to interaction mode autonomously.

The manipulation task idealized in the beginning of this study was configured with a force/torque sensor mounted on the interface between the object on the wall and the gripper. The equilibrium-based force/torque controller was tested with a spring-mounted device. Experiments with a force/torque sensor are still recommended to validate the controller for the regulation of the six-dimensional force/torque vector.

The computation of the dynamic equation terms of an aerial manipulator requires high processing capability. Several partial derivatives are required to compute the terms associated with the centrifugal and Coriolis forces. In this research, the aerial manipulator was not required to perform high-speed maneuvers. For this reason, some terms of the dynamic equations could be neglected to reduce computational load. The embedded controller implemented in this study worked appropriately with the microcontroller Pixhawk. If for other experiments an aerial manipulator is required to perform high-speed maneuvers, all the terms in the dynamic equation are significant. As a future work, it is recommended to study if the microcontroller Pixhawk is capable of running a more complex controller, taking into consideration all the terms of the dynamic equation. If this is not the case, it is suggested to integrate another embedded computer into the aerial manipulator to perform the required computations in real time.

# LIST OF REFERENCES

[1] V. Lippiello and F. Ruggiero, “Cartesian impedance control of a UAV,” in *10th IFAC Symposium on Robot Control*, Dubrovnik, Croatia, 2012, pp. 704–709.

[2] V. Lippiello and F. Ruggiero, “Exploiting redundancy in Cartesian impedance control of UAVs equipped with a robotic arm,” in *International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, 2012, pp. 3768–3773.

[3] F. Caccavale, G. Giglio, G. Muscio, and F. Pierri, “Adaptive control for UAVs equipped with a robotic arm,” in *World Congress The International Federation of Automatic Control*, Cape Town, South Africa, 2014, pp. 11049–11054.

[4] G. Arleo, F. Caccavale, and F. Pieri, “Control of quadrotor aerial vehicles equipped with a robotic arm,” in *Mediterranean Conference on Control & Automation*, Platanias-Chania, Crete, Greece, 2013, pp. 1174–1180.

[5] G. Giglio and F. Pierri, “Selective compliance control for an unmanned aerial vehicle with a robotic arm,” in *Mediterranean Conference on Control and Automation*, Palermo, Italy, 2014, pp. 1190–1195.

[6] M. Orsag, C. Korpela, and P. Oh, “Modeling and control of MM-UAV: mobile manipulating unmanned aerial vehicle,” *J. of Intell. and Robot. Syst.*, vol. 69, no. 1, pp. 227–240, Jan. 2013.

[7] T. W. Danko and P. Y. Oh, “Design and control of a hyper-redundant manipulator for mobile manipulating unmanned aerial vehicles,” *J. of Intell. and Robot. Syst.*, vol. 73, no. 1, pp. 709–723, Jan. 2014.

[8] C. Korpela, M. Orsag, and P. Oh, “Hardware-in-the-loop verification for mobile manipulating unmanned aerial vehicles,” *J. of Intell. and Robot. Syst.*, vol. 73, no. 1, pp. 725–736, Jan. 2014.

[9] T. W. Danko and P. Y. Oh, “Evaluation of visual servoing control of aerial manipulators using test gantry emulation,” in *International Conference on Unmanned Aircraft Systems (ICUAS)*, Orlando, FL, 2014, pp. 821–829.

[10] M. Orsag and C. M. Korpela, “Hybrid adaptive control for aerial manipulation,” *J. of Intell. and Robot. Syst.*, vol. 73, no. 1, pp. 693–707, Jan. 2014.

[11] M. Orsag, C. Korpela, S. Bogdan, and P. Oh, “Valve turning using a dual-arm aerial manipulator,” in *International Conference on Unmanned Aircraft Systems*, Orlando, FL, 2014, pp. 836–841.

[12] A. Jimenez-Cano, J. Martin, G. Heredia, A. Ollero, and R. Cano, "Control of an aerial robot with multi-link arm for assembly tasks," in *IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013, pp. 4916–4921.

[13] G. Heredia, A. Jimenez-Cano, I. Sanchez, D. Llorente, V. Vega, J. Braga, J. Acosta, and A. Ollero, "Control of a multirotor outdoor aerial manipulator," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, IL, 2014, pp. 3417–3422.

[14] J. Acosta, M. Sanchez and A. Ollero, "Robust control of underactuated aerial manipulators via IDA-PBC," in *IEEE Conference on Decision and Control*, Los Angeles, California, 2014, pp. 673–678.

[15] F. Ruggiero, J. Cacace, H. Sadeghian, and V. Lippiello, "Impedance control of VToL UAVs with a momentum-based external generalized forces estimator," in *International Conference on Robotics & Automation*, Hong Kong, China, 2014, pp. 2093–2099.

[16] F. Ruggiero, M. Trujillo, R. Cano, H. Ascorbe, A. Viguria, C. Perez, V. Lippiello, A. Ollero, and B. Siciliano, "A multilayer control for multirotor UAVs equipped with a servo robot arm," in *International Conference on Robotics & Automation*, Seattle, Washington, 2015, pp. 4014–4020.

[17] H. Yang and D. Lee, "Dynamics and control of quadrotor with robotic Manipulator," in *International Conference on Robotics & Automation*, Hong Kong, China, 2014, pp. 5544–5549.

[18] S. Kim, S. Choi, and H. J. Kim, "Aerial manipulation using a quadrotor with a two DOF robotic arm," in *International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 4990–4995.

[19] G. Darivianakis, K. Alexis, M. Burri, and R. Siegwart, "Hybrid predictive control for aerial robotic physical interaction towards inspection operations," in *International Conference on Robotics & Automation*, Hong Kong, China, 2014, pp. 53–58.

[20] K. Alexis, G. Darivianakis, M. Burri, and R. Siegwart, "Aerial robotic contact-based inspection: planning and control," *Auton. Robot.*, vol. 40, no. 4, pp. 631–655, Apr. 2016.

[21] C. Papachristos, K. Alexis, and A. Tzes, "Efficient force exertion for aerial robotic manipulation: exploiting the thrust–vectoring authority of a tri–tiltrotor UAV," in *IEEE International Conference on Robotics & Automation*, Hong Kong, China, 2014, pp. 4500–4505.

[22] D. Lee and C. Ha, "Mechanics and control of quadrotors for tool operation," in *Annual Dynamic Systems and Control Conference*, Fort Lauderdale, Florida, 2012, pp. 177–184.

[23] T. Tomic and S. Haddadin, “A unified framework for external wrench estimation, interaction control and collision reflexes for flying robots,” in *International Conference on Intelligent Robots and Systems*, Chicago, IL, 2014, pp. 4197–4204.

[24] J. L. Scholten, M. Fumagalli, S. Stramigioli, and R. Carloni, “Interaction control of an UAV endowed with a manipulator,” in *International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013, pp. 4910–4915.

[25] M. Fumagalli and R. Carloni, “A modified impedance control for physical interaction of UAVs,” in *International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 1979–1984.

[26] A. Y. Mersha, S. Stramigioli, and R. Carloni, “Exploiting the dynamics of a robotic manipulator for control of UAVs,” in *International Conference on Robotics and Automation*, Hong Kong, China, 2014, pp. 1741–1746.

[27] M. Fumagalli, R. Naldi, A. Macchelli, R. Carloni, S. Stramigioli, and L. Marconi, “Modeling and control of a flying robot for contact inspection,” in *International Conference on Intelligent Robots and Systems*, Vilamoura, Algarve, Portugal, 2012, pp. 3532–3537.

[28] A. Y. Mersha, S. Stramigioli, and R. Carloni, “Variable impedance control for aerial interaction,” in *International Conference on Intelligent Robots and Systems*, Chicago, IL, 2014, pp. 3435–3440.

[29] G. Jiang and R. Voyles, “Hexrotor UAV platform enabling dextrous interaction with structures-flight test,” in *International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Linkoping, Sweden, 2013, pp. 1–6.

[30] D. C. Robinson, H. Chung, and K. Ryan, “Computational investigation of micro rotorcraft near-wall hovering aerodynamics,” in *International Conference on Unmanned Aircraft Systems*, Orlando, FL, 2014, pp. 1055–1063.

[31] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics Modelling, Planning and Control*, London: Springer-Verlag, 2010.

[32] B. Wie, *Space Vehicle Dynamics and Control*, 2nd ed. Reston, VA: American Institute of Aeronautics and Astronautics, 2008.

[33] E. Capello, H. Park, B. Tavora, G. Guglieri, and M. Romano, “Modeling and experimental parameter identification of a multicopter via a compound pendulum test rig,” in *Workshop on Research, Education and Development of Unmanned Aerial Systems*, Cancun, Mexico, 2015, pp. 308–317.

[34] S. Bouabdallah and R. Siegwart, “Full control of a quadrotor,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, 2007, pp. 153–158.

[35] G. Antonelli, E. Cataldi, P. R. Giordano, S. Chiaverini, and A. Franchi, “Experimental validation of a new adaptive control scheme for quadrotors MAVs,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 2439–2444.

[36] H. Huang, G. M. Hoffmann, S. L. Waslander, and C. J. Tomlin, “Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering,” in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, 2009, pp. 3277–3282.

# INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California